Application of Convex Optimization Techniques for Feature Extraction from EEG Signals

by

Zahra Roshan Zamir

A thesis submitted in fulfilment of the
requirements for the degree of
Doctor of Philosophy

Department of Mathematics
Faculty of Science, Engineering and Technology
Swinburne University of Technology
Melbourne, Australia

2016

Abstract

Electroencephalogram (EEG) is an electrical activity of the human brain that can be recorded graphically. The analysis of EEG recordings is important for extracting the relevant information from the human brain activities. In addition, this analysis is significant for detection of unwanted transient events in order to diagnose brain diseases. The visual screening of long term EEG recordings is a time consuming and difficult task, and it is insufficient for capturing reliable information from brain activities. Therefore an automatic screening of EEG recordings is required to diagnose brain diseases. In particular, EEG recordings are important in the identification of sleep stages and in the diagnosis of epilepsy. Furthermore, the classification of EEG signals in presence of transient events such as K-complexes and seizures is essential to diagnose people who suffer from sleep disorder and epilepsy respectively. K-complex is a special type of EEG waveform that is used in the sleep stage scoring. An automated detection of K-complexes is a desirable component of the sleep stage monitoring. This automation is difficult due to the ambiguity of the scoring rules, stochastic nature of brain signals, presence of noise, complexity and extreme size of data. Many research efforts have been focused on the K-complex detection via several methods such as neural networks, wavelet transform and non-smooth optimization. However, drawbacks include false classification of a severe noise or a sleep spindle as a K-complex and the long computational time. While some attempts were made to develop algorithms to detect K-complexes, the reported success rates are subject to change from person to person and no standard algorithm has been accepted by the medical community. An epileptic seizure is a transient event of abnormal excessive neuronal discharge in the brain. This unwanted event can be obstructed by detection of electrical changes in the brain signal that happen before the seizure takes place. The automatic detection of seizures is necessary since the visual screening of EEG recordings is a time consuming task and requires experts to improve the diagnosis of epilepsy. There have been various attempts for automatic detection of epilepsy based on artificial neural networks, genetic programmings and wavelet transforms. However, more research should be conducted for automatic detection of seizures since it is a challenging task due to inconsistency of signals in patients. Moreover, none of the previous methods was investigated the consistency in performances. To address these issues, four convex optimization models are developed to extract key features of EEG signals. The first two models are linear least squares problems while the last two models are uniform approximation problems. These models are based on the approximation of original signals by sine functions. The signal amplitude is approximated by a piecewise polynomial and a polynomial functions. Thereafter, the parameters of the corresponding approximations (rather than raw data) are used to detect K-complexes and seizures. The proposed approach significantly reduces the dimension of the classification problem (by extracting essential features) and the computational time while the classification accuracy is improved. The linear least squares problems have been investigated in order to analyze the singularity of their system matrices. If the system matrix is non-singular, then, the corresponding problem can be solved inexpensively and efficiently, while for singular cases, slower (but more robust) methods have to be used. To choose a better suited method for solving the corresponding linear least squares problems, the singularity verification rules have been developed. This thesis develops necessary and sufficient conditions for non-singularity of first model, and sufficient conditions for non-singularity and singularity of second model. Therefore the algorithm efficiency can be improved by choosing a suitable method for solving the corresponding linear least squares problems. In addition, the uniform approximation problems have been reformulated as linear programming problems. Numerical results show that four convex optimization models involving piecewise polynomials are efficient for detecting K-complexes. The first model involving a piecewise polynomial and a polynomial as amplitudes is a promising feature extraction method for detecting epileptic seizures.

Dedicated to my husband, Masoud Goudarzi who has been an endless source of love and encouragement. There is no doubt in my mind that without his continued moral support I could not finish this doctoral journey. This thesis is also dedicated to my parents, Soghra and Karim for their endless love, support and for believing in me at what I am doing.

Acknowledgment

In preparing this thesis, I have been in contact with various researchers, academicians and practitioners. They have contributed to my knowledge and understandings of this project. At first I wish to express my deepest gratitude to my principal supervisor, Dr. Nadezda Sukhorukova and co-supervisor, Associate Prof. Sergey Suslov, for their support, patience, motivation, guidance and immense knowledge. Without their suggestions, helps and criticisms, this thesis would not been as it is presented now.

I wish to extend my gratitude to Swinburne University of Technology for giving me the opportunity to do this research and for granting me Postgraduate Research Award (SUPRA). I am also indebted to all academicians in Department of Mathematics for their comments, support and discussions. I am most grateful to Prof. Billy Todd, Head of Department of Mathematics for his support and helps during my PhD research.

Many questions concerning my future academic endeavors are answered in the words of Sir Winston Churchill, "*Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning*."

Last, but by no means least, I would like to thank to all my family members for always been supportive during all my endeavors. I am also deeply appreciated to my mother, father and my beloved husband, Masoud for their spiritual support, patience and love.

Zahra Roshan Zamir, 2016

Declaration

I declare that this thesis entitled "Application of Convex Optimization Techniques for Feature Extraction from EEG signals" is the result of my own research except as cited in the references. To the best of my knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other candidate for any other degree or diploma, except where due reference is made in the text of the thesis.

Zahra Roshan Zamir
November 2016

Publications

Published papers

Z. Roshan Zamir. Detection of epileptic seizure in EEG signals using linear least squares preprocessing, Computer Methods and Programs in Biomedicine, Volume 133, Pages 95–109, August 2016. http://dx.doi.org/10.1016/j.cmpb.2016.05.002. http://www.sciencedirect.com/science/article/pii/S016926071530273X.

Z. Roshan Zamir and N. Sukhorukova. Linear least squares problems involving fixed knots polynomial splines and their singularity study, Applied Mathematics and Computation, Volume 282, Pages 204–215, May 2016. http://dx.doi.org/10.1016/j.amc.2016.02.011.

Z. Roshan Zamir, N. Sukhorukova, H. Amiel, A. Ugon and C. Philippe. Convex optimisation-based methods for K-complex detection, Applied Mathematics and Computation, Volume 268, Pages 947–956, July 2015. http://www.sciencedirect.com/science/article/pii/S0096300315009169.

Z. Roshan Zamir, N. Sukhorukova, H. Amiel, A. Ugon and C. Philippe. Optimization-based features extraction for K-complex detection, In Mark Nelson, Tara Hamilton, Michael Jennings, and Judith Bunder, editors, Proceedings of the 11th Biennial Engineering Mathematics and Applications Conference, EMAC-2013, volume 55 of ANZIAM J., pages C384–C398, August 2014. http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/7802 [August 27, 2014].

Conference abstracts

Z. Roshan Zamir and N. Sukhorukova. Characterizing an EEG signal through linear least squares and convex optimization modelings. South Pacific Continuous Optimization Meeting (SPCOM 2015), 08–12 February, 2015, University of South Australia, Adelaide, SA, Australia.

Z. Roshan Zamir and N. Sukhorukova. Characterizing an EEG signal through linear least squares and convex optimization modelings. The 17th biennial Computational Techniques and Applications Conference (CTAC), 1–3 December, 2014, Australian National University (ANU), Canberra, ACT, Australia.

Z. Roshan Zamir and N. Sukhorukova. Optimization-based features extraction for K-complex detection. Constructive Optimization Workshop, 16–17 April, 2014, Federation University Australia (former University of Ballarat), Ballarat, Victoria, Australia.

Z. Roshan Zamir and N. Sukhorukova. Optimization-based features extraction for K-complex detection. 11th Engineering Mathematics and Applications Conference (EMAC), 1–4 December, 2013, Queensland University of Technology (QUT), Brisbane, Queensland, Australia.

Contents

List of Figures

List of Tables

List of Abbreviations

AASM	–	American Academy of Sleep Medicine
ACC	–	Classification Accuracy
ANNs	–	Artificial Neural Networks
CGP	–	Constructive Genetic Programming
CMs	–	Confusion Matrices
ECG	–	Electrocardiogram
EEG	–	Electroencephalography
EMD	–	Empirical Mode Decomposition
EMG	–	Electromyogram
EOG	–	Electrooculogram
FFT	–	Fast Fourier Transform
FNR	–	False Negative Rate
FP	–	False Positive
FPR	–	False Positive Rate
GP	–	Genetic Programming
IMFs	–	Intrinsic Mode Functions
IPM	–	Interior Point Method
LASSO	–	Least Absolute Shrinkage and Selection Operator
LS-SVM	–	Least Square SVM
LLSOM1	–	Linear Least Squares Optimization Model 1
LLSOM2	–	Linear Least Squares Optimization Model 2
LLSOM1–2	–	LLSOM1 and LLSOM2
LLSP	–	Linear Least Squares Problem
LLSP1	–	LLSOM1 involving a polynomial as an amplitude
LLSP2	–	LLSOM2 involving a polynomial as an amplitude

LLSP3	–	LLSOM1 involving a spline as an amplitude
LLSP4	–	LLSOM2 involving a spline as an amplitude
LLSP1– 4	–	LLSP1, LLSP2, LLSP3 and LLSP4
LPP	–	Linear Programming Problem
LSP	–	Least Squares Problem
MFK	–	Mean Frequency for K-complexes
MFNK	–	Mean Frequency for non-K-complexes
MOF	–	Mean of Objective Function values
MRI	–	Magnetic Resonance Imaging
NLSPs	–	Non-linear Least Squares Problems
NREM	–	Non-Rapid Eye Movement
OPAs	–	Optimization-based Preprocessing Approaches
OPAs	–	LLSOM1, LLSOM2, UOM1 and UOM2 in Chapter 5
OPAs	–	LLSP1, LLSP2, LLSP3 and LLSP4 in Chapter 6
PSG	–	Polysomnogram
RBF	–	Radial Basis Function
REM	–	Rapid Eye Movement
RIC	–	Representation Instance-based Classification
R & K	–	Rechtschafen and Kales
ROC	–	Receiver Operating Characteristic
SM	–	Simplex Method
SODP	–	Second-order Differential Plot
STFT	–	Short-Time Fourier Transform
SVD	–	Singular Value Decomposition
SVM	–	Support Vector Machine
SWS	–	Slow-Wave-Sleep
TMF	–	Threshold value for Mean Frequency
TP	–	True Positive
TPR	–	True Positive Rate
UOM1	–	Uniform Optimization Model 1
UOM2	–	Uniform Optimization Model 2
UOM1–2	–	UOM1 and UOM2
WLS	–	Weighted Least Squares

List of Symbols in Order of Appearance

$f_0(x)$	–	Objective function
$x \in \mathbb{R}^n$	–	Decision or Optimization variable
f_i, h_k	–	Inequality and equality constraint functions respectively
x^*	–	A local minimum or a global minimum
$\delta(x^*)$	–	A neighborhood of a local minimum
Tr	–	A given training set
y_i	–	EEG recordings at time t_i
N	–	Total number of recordings in a time segment $t = [t_1, t_N]$
$A(t_i)$	–	Signal Amplitude
ω	–	Sampling frequency
τ	–	Phase shift
P	–	Polynomial function
m_2	–	Degree of a polynomial
$\mathbf{x}$	–	$\mathbf{x} \in \mathbb{R}^{m_2+1}$ are polynomial parameters
$\mathbf{x}$	–	$\mathbf{x} = (x_0, x_1, \ldots, x_{m_2}) \in \mathbb{R}^{m_2+1}$
S	–	Piecewise polynomial (spline) function
m_1	–	Degree of a spline
n	–	Number of subintervals
$\mathbf{x}$	–	$\mathbf{x} = (x_0, x_{11}, \ldots, x_{m_1 n}) \in \mathbb{R}^{m_1 n+1}$ are spline parameters
D	–	Duration of an EEG signal in seconds
$\boldsymbol{\theta}$	–	$\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n-1})$ are spline knots
W_1	–	Signal approximation (modulation)
$\boldsymbol{M}$	–	The system matrix obtained from LLSOM1

$\mathbf{y}$	–	$\mathbf{y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ is a signal segment
W_2	–	Signal approximation by shifting W_1 vertically
$\boldsymbol{B}$	–	System matrix obtained from LLSOM2, $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$
$\boldsymbol{U}$	–	$N \times N$ orthogonal matrix
$\boldsymbol{V}$	–	$(2m_1n + 2) \times (2m_1n + 2)$ orthogonal matrix
$\boldsymbol{\Sigma}$	–	$N \times (2m_1n + 2)$ diagonal matrix
σ_{ij}	–	The diagonal entries of $\boldsymbol{\Sigma}$
σ_i	–	Singular values of $\boldsymbol{B}$
$\boldsymbol{V}$	–	$(2m_1n + 2) \times (2m_1n + 2)$ orthogonal matrix
ω_i, ω_f	–	Initial and final values for ω
τ_i, τ_f	–	Initial and final values for τ
$\boldsymbol{E}$	–	System matrix obtained from LLSOM1 or LLSOM2
a	–	Entry $\{11\}$ of a confusion matrix
b	–	Entry $\{12\}$ of a confusion matrix
c	–	Entry $\{21\}$ of a confusion matrix
d	–	Entry $\{22\}$ of a confusion matrix
A	–	EEG recordings of healthy volunteers with eyes open
B	–	EEG recordings of healthy volunteers with eyes close
C, D	–	EEG recordings of patients during seizure free intervals
E	–	EEG recordings of patients during seizure activities
S_1, S_2	–	Splines with fixed knots
$g(t)$	–	A prototype function
$f(t)$	–	A modeling function
$\boldsymbol{D}$	–	A diagonal matrix with $g(t_i)$ diagonal entries $\boldsymbol{M} = \boldsymbol{D}\boldsymbol{V}$
i	–	$i = 1, 2, \ldots, N$
$\boldsymbol{V}$	–	A Vandermonde type matrix
$\widetilde{\boldsymbol{B}}$	–	A matrix obtained by rearranging the columns of $\boldsymbol{B}$
$\boldsymbol{A}$	–	A block lower triangular matrix made from $\boldsymbol{M}$ and $\boldsymbol{B}$
$\boldsymbol{A}_{ii}$	–	Diagonal entries (sub-blocks) of matrix $\boldsymbol{A}$
i	–	$i = 1, 2, \ldots, n$
N_l	–	The total number of recordings in $l-$th subinterval
l	–	$l = 1, 2, \ldots, n$
K_l	–	The number of time moments t_i in the $l-$th subinterval
$\boldsymbol{\Lambda}_{11}$	–	First sub-block of $\boldsymbol{A}_{11}$
$\boldsymbol{\Lambda}_{12}$	–	Second sub-block of $\boldsymbol{A}_{11}$

$\boldsymbol{\Lambda}_{21}$	–	Third sub-block of $\boldsymbol{A}_{11}$
$\boldsymbol{\Lambda}_{22}$	–	Fourth sub-block of $\boldsymbol{A}_{11}$
$\boldsymbol{\Lambda}_{12}^{j}$	–	The $j-$th row of $\boldsymbol{\Lambda}_{12}$, $j = 1, \ldots, N_1 - m - 1$
$\boldsymbol{\Lambda}_{22}^{i}$	–	The $i-$th row of $\boldsymbol{\Lambda}_{22}$, $i = 1, \ldots, m + 1$
$\boldsymbol{\Lambda}_{11}^{j}$	–	The rows of $\boldsymbol{\Lambda}_{11}$
$\boldsymbol{\Lambda}_{21}^{i}$	–	The rows of $\boldsymbol{\Lambda}_{21}$
$\tilde{\boldsymbol{\Lambda}}_{11}$	–	An updated sub-block obtained from $\boldsymbol{\Lambda}_{11}$
$\tilde{\boldsymbol{\Lambda}}_{11}^{j}$	–	The rows of the updated sub-block $\tilde{\boldsymbol{\Lambda}}_{11}$
K_l	–	The multiplicity coefficients for $l-$th subinterval
$\widetilde{K}$	–	The multiplicity coefficient for a given subinterval
$\widetilde{N}$	–	The number of recordings in each subinterval
$\boldsymbol{A}_d$	–	Each diagonal block of matrix $\boldsymbol{A}$
$\widetilde{\boldsymbol{A}}_d$	–	A matrix obtained by rearranging the time moments
$\boldsymbol{A}_1$, $\boldsymbol{A}_2$	–	Form matrix $\widetilde{\boldsymbol{A}}_d$
$\tilde{\tilde{\boldsymbol{A}}}_d$	–	A matrix obtained by a linear transformation on $[\boldsymbol{A}_1\ \boldsymbol{A}_2]$
Z	–	A new variable to make a function linear
$\boldsymbol{e}$	–	A matrix of ones
$\boldsymbol{Q}$	–	A block matrix containing $\boldsymbol{M}$, $-\boldsymbol{M}$ and $-\boldsymbol{e}$
$\mathbf{X}$	–	A vector contains a vector of $\mathbf{x}$ and variable Z
$\mathbf{Y}$	–	A vector contains y_i and $-y_i$
$\boldsymbol{E}$	–	A block matrix containing $\boldsymbol{B}$, $-\boldsymbol{B}$ and $-\boldsymbol{e}$

Chapter 1

Introduction

1.1 Introduction

The structure of the human brain is complex and its functions are diverse. The brain is the center of the nervous system and it controls all physical and mental functions including muscle movements, breathing, walking, memory, thinking, learning, emotion, attention and so on. To control body actions, such as watching television, the brain generates electrical signals through neurons that exploit chemical reactions. There exist scientific techniques to monitor these electrical signals, for instance, electroencephalography (EEG) and magnetic resonance imaging (MRI).

Any intrusion into the human brain, such as surgery to monitor its functions, can damage it. Therefore a non invasive technique is necessary in order to capture its electrical activity. EEG has become a widely used technique to record the electrical activity of the human brain because of its excellent properties including non-invasiveness, millisecond-range temporal resolution and low cost. EEG is an important technique to diagnose brain diseases and abnormalities such as sleep disorders and epilepsy that cause electrical changes in the brain.

Classification of EEG signals through partitioning EEG segments is necessary to decide whether volunteers are healthy or not. EEG generates complex and large scale data and visual screening of long term EEG recordings is a time consuming and difficult task and it is insufficient for capturing reliable information. Therefore development of automatic methods for detection of unwanted transient events in EEGs and extraction of information from a large amount of data is vital. This thesis considers two important medical applications that are based on the classification of EEG signals: epileptic seizure detection and sleep stage identification. Four methods have been developed to automate detection of transient events, which helps the practitioners identify sleep stages and diagnose seizures. Twelve classification algorithms have been employed to classify EEG signals in the presence of a transient event.

The proposed methods extract the essential features of EEG signals that describe the characteristics of the original signal with sufficient accuracy. Consequently, they reduce the dimensionality of complex data obtained from EEG recordings. The proposed methods may be useful for sleep disorder specialists and neurologists to identify some brain diseases efficiently using the typical patterns of EEG signals. The outcomes of this research can facilitate better care services for patients with sleep disorders and seizures while saving time and reducing costs.

Optimization methods are important tools in the fields of signal processing and data fitting. Medical practitioners, for example, want to reduce the time required for diagnosis and maximize the accuracy of diagnosis. Because the time required and the accuracy achieved in any practical situation can be modeled as a function of decision variables, optimization can be used efficiently to detect transient events in EEG signals within a time limit.

Most real world problems have many decision variables; therefore they have a large number of local minima and maxima that might not be the global solutions. Although finding the local optimum is simple using local optimization methods, finding the global optimum is a difficult task. One of the main drawbacks of any local optimization method is that it does not search the whole domain of the objective function to reach the global optimum.

Formulating a real life problem as a non-convex optimization problem is straightforward, but solving it using local optimization methods is difficult. Although formulating and recognizing a practical problem as convex optimization is a challenging task, there exist many reliable algorithms to solve them. One of the main properties of convex optimization problems is that any locally optimal solution is also globally optimal. Taken together, the art in a non-convex optimization problem is in solving it whereas the art in a convex optimization problem is in formulating it. Therefore this research is based on mathematical modeling of EEG signals captured from human brains through convex optimization for automatic detection of transient EEG waveforms. This study can be considered an application of convex optimization techniques in biomedical signal processing.

1.2 Background of the Problem

An EEG is a graphical recording of electrical activity of the human brain. These signals provide significant contributions in biomedical signal processing. The assessment of EEG signals is essential for extracting the relevant information from EEG recordings in order to diagnose brain diseases and abnormalities. Two important applications of EEG signals are sleep stage identification and epileptic seizure detection.

K-complex is a special type of transient EEG waveform that is used in sleep stage identification. In manual identification, medical doctors look for specific patterns (sleep events) based on specially developed scoring rules [41]. Then, one of the sleep stages (awake, sleep stage 1, 2, 3 or REM) is assigned to each signal segment of an EEG (called an epoch, normally lasting 30 seconds). This is a time consuming task. Therefore an accurate method for automation of manual identification is desirable. This automation is difficult, due to the stochastic nature of brain signals, the presence of noise, complexity and extreme size of data. While a few studies have been carried out to automate detection of K-complexes, no standard algorithm has been accepted by the medical community. Consequently, this study focuses on an automatic detection of K-complexes that are essential waveforms for distinguishing between sleep stage 1 and sleep stage 2 of an EEG signal.

An epileptic seizure is a transient event of abnormal excessive neuronal discharge in the brain. This unwanted event can be obstructed by detection of electrical changes in the brain that happen before the seizure takes place. Hence, the automatic detection of seizures is necessary since the visual screening of EEG recordings is a time consuming task and requires experts to improve the diagnosis. Although there have been studies in automatic detection of epileptic seizure, more research is required, due to inconsistency of signals in relation to patients' sex, age and so on.

To address these, this study has developed methods for automatic detection of transient EEG waveforms called K-complex and epileptic seizure in order to extract the key features of EEG signals and improve the performance of classification algorithms.

1.3 Statement of the Problem

This research is aimed at investigating the efficiency and effectiveness of optimization methods and developing optimization models for automatic detection of K-complexes and epileptic seizures for sleep staging of an EEG and seizures diagnosis respectively.

One of the main characteristics of K-complexes and epileptic seizures is a sudden and excessive increase in signal amplitude. Continuous piecewise polynomials that are known as splines are flexible candidates to model abrupt changes in amplitude. In addition, it seems that simpler modeling functions such as polynomials work well. Therefore these facts motivate us to model the brain signal (EEG) as sinusoidal waves with the signal amplitude approximated by a spline and a polynomial (which is simpler than a spline) [63]. These enable one to

- develop an accurate model for the wave shapes and
- extract key features of the waves that are crucial for detecting K-complexes and seizures.

These features are extracted through minimizing

- the sum of the squares of the deviation between the original signal and the modeling functions;
- the maximum of the absolute deviation between the original signal and the approximated wave.

The first approach necessitates solving a sequence of linear least squares problems that is a subclass of convex optimization problems. The second approach necessitates solving a convex optimization problem based on uniform (Chebyshev) approximation.

After extracting the features, classification algorithms (classifiers) are applied over the set of extracted features to evaluate the classification accuracy of an EEG signal in the presence of K-complexes and seizures.

1.4 Aims and Objectives of the Study

The major goals of this study are to:

- investigate the possibility of using optimization methods in EEG signal processing;
- analyze how the existing methods can be improved for more accurate and efficient classification of EEG signals.

The objectives of this study are to:

- develop convex optimization-based procedures for extracting a set of signal characteristics that is essential for K-complex and seizure detection;
- formulate the feature extraction procedure as an optimization problem;
- identify relevant optimization methods to solve the problem;
- identify efficient classification algorithms based on the classification accuracy and computational time;
- improve the current procedure of EEG signal classification in presence of K-complex and seizure;
- run the numerical tests and carry out simulations using the developed procedures.

1.5 Contributions

New convex optimization-based models were developed for automatic detection of K-complexes and seizures. The procedure consists of two major steps.

1. We extracted key features of an EEG signal and reduced the dimensions of the data. This procedure was based on convex optimization.

2. We applied classification algorithms to evaluate the classification accuracy of an EEG signal in the presence of K-complexes and seizures.

There are two main contributions of this work:

1. The development of optimization models that are
   - categorized under convex optimization problems;
   - simple and computationally inexpensive to solve;
   - accurate enough to achieve a high level of classification accuracy with less execution time.
2. The implementation and comparison of the proposed models, in particular, their classification accuracy and the corresponding computational time.

1.6 Organization of the Thesis

This thesis is presented into seven chapters. The content of each chapter is organized as follows.

Chapter 1 gives the introduction of this study. This chapter discussed the introduction of the research area, background of the problem, statement of the problem, aims and objectives of the study, and contributions of the study.

Chapter 2 reviews the optimization area, biomedical signal processing, classification and feature extraction of EEG signals, and solution methods containing linear programming, linear least squares and convex optimization methods.

Chapter 3 introduces novel methods based on the sequence of linear least squares and uniform approximation problems. These are developed to extract the essential features of EEG signals in order to detect K-complexes and seizures automatically and to enhance the performance of classification algorithms.

Chapter 4 provides an extensive mathematical analysis of linear least squares and uniform approximation models. Singularity verification rules are developed for linear least squares models in order to choose a better suited method for solving them. It is demonstrated that uniform approximation models can be formulated as linear programming problems. This formulation can be used to verify that the solutions are optimal.

Chapter 5 illustrates the efficiency and robustness of developed models in automatic detection of K-complexes. It reports the numerical results obtained from proposed methods when an EEG signal is formulated as a sine wave and its amplitude is approximated as a spline and polynomial. Further, different classification algorithms are applied to evaluate the performance of developed methods in order to enhance the classification accuracy. This chapter discusses three issues as follows.

- Which optimization-based model is more robust and efficient in the extraction of key features of an EEG signal in presence of K-complexes?
- Which amplitude approximation (spline or polynomial) is the best for this particular dataset?
- Which combination of optimization-based models and classification algorithms is best in terms of classification accuracy?

Chapter 6 reports numerical results obtained from employing linear least squares-based methods over an EEG epileptic dataset. This EEG signal is modeled as a sine wave where its amplitude is approximated as a spline and polynomial. Different classification algorithms are applied in order to investigate the improvements in classification accuracy that is achieved through employing linear least squares-based methods. This chapter discusses three issues as follows.

- Which method is more robust and efficient in the extraction of essential features of an EEG signal in presence of seizures?
- Which amplitude modulation is the best for an EEG epileptic dataset?
- Which combination of linear least squares-based methods and classification algorithms is the best in terms of classification accuracy?

Chapter 7 provides a summary and conclusions of this research. In addition, this chapter includes suggestions for further studies.

Chapter 2

A review on Optimization and Its Applications in EEG Signal Classification

2.1 Introduction

This chapter contains an overview of optimization with particular focus on convex optimization, biomedical signal processing, feature extraction and classification of different categories of EEG signals including their concepts and structures. The goal of this thesis is to develop convex optimization-based methods to extract the essential features of an EEG signal in order to approximate the EEG signal accurately, enhance the performance of classification algorithms and reduce the dimensionality of EEG datasets.

2.2 Optimization

Optimization is the science of identifying the best solutions to certain mathematical problems that model real life situations. It may be defined traditionally as an aid to human decision making. In our daily lives, we make many technological and managerial decisions. It is widely applied in engineering design, science, management, economics and business. For example, an engineer is in the decision loop, continually reviewing the proposed designs and changing parameters in the problem specification if needed.

In most situations, the concept of a best decision depends on a problem considered and is not easy to define mathematically. The most common way of doing this is to represent the decision as a set of parameters called *decision variables*. Most practical situations can be modeled as an objective function of decision variables subject to constraints. Then, optimization can be defined as finding the best solution within a given time limit, which is concerned with finding the maximum or minimum value of this objective function depending on decision variables.

One of the difficulties in the optimization is to find an effective method to estimate an optimizer for a real valued function with n variables, especially if n is large. Once the model is formulated mathematically, an optimization algorithm is used to solve it. Generally, optimization problem requires a computer to complete this process since the algorithms and models are complicated and time consuming [61]. The basic components of optimization problems are:

- a set of decision variables,
- an objective function to be minimized (cost function) or maximized (profit function) and
- a set of constraints that specify feasible values of the decision variables.

Optimization problems involve finding the values of the decision variables that optimize the objective function while satisfying the constraints. In optimization problems, decision variables can admit continuous, discrete or symbolic values. A problem is continuous if the decision variables take on continuous real values, otherwise, it is a discrete problem. The decision variables of a discrete problem usually admit integer values. The constraints can also have linear or non-linear forms and be defined implicitly or explicitly or may not even exist. They can be equalities or inequalities. Optimization problems may be constrained or unconstrained depending on the existence of conditions imposed on the decision variables.

A general constrained minimization problem can be formulated as follows:

$$\text{minimize } (f_0(x)) \quad \text{subject to} \quad \begin{cases} f_i(x) \leq 0\,, & i = 1, \ldots, m\,, \\ h_k(x) = 0\,, & k = 1, \ldots, p\,, \end{cases} \tag{2.1}$$

where $x \in \mathbb{R}^n$ is the decision variable (optimization variable), function $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function, and functions $f_i, h_k : \mathbb{R}^n \to \mathbb{R}$ are inequality and equality constraint functions respectively. Maximization problems are transformed into minimization problems such that in (2.1)

$$\max\left(f_0(x)\right) = -\min\left(-f_0(x)\right).$$

Optimization problems can be divided into several classes.

1. Classification of optimization problems based on the existence of constraints.
   - ***Constrained Optimization Problems***: problems that are subject to one or more constraints.
   - ***Unconstrained Optimization Problems***: if $m = p = 0$ in problem (2.1) then, it is called *unconstrained optimization problem*.
2. Classification of optimization problems based on the type of objective function and constraints.
   - ***Linear Optimization Problems***: the objective function and the constraints are linear.
   - ***Non-linear Optimization Problems***: if either of objective and constraint functions, or both, are non-linear.
   - ***Convex Optimization Problems***: the objective and the inequality constraint functions in (2.1) are convex and the equality constraint functions are affine. A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is affine if it has the form $f(x) = Ax + b$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.
   - ***Non-convex Optimization Problems***: the objective function and some or all the constraints may be non-convex functions.
3. Classification of optimization problems based on the type of solution obtained.
   - ***Local Optimization Problems***: a vector $x^* \in \Omega$ is said to be a local minimum of $f : \Omega \to \mathbb{R}$ if there exists a neighborhood $\delta(x^*) : f(x^*) \leq f(x)\ \forall x \in \delta(x^*)$. A major drawback of any local optimization algorithm is that it does not search the whole domain which is

evidently necessary in global optimization.

- ***Global Optimization Problems***: a vector $x^* \in \Omega$ is said to be a global minimum if $f(x^*) \leq f(x)\ \forall x \in \Omega$. The aim of global optimization is to find the solution, for which the objective function achieves its optimal value in the complex region of interest.

There exist various type of optimization problems, so, there is no specific method for solving them since each problem has special requirements to obtain its optimal solutions.

2.3 Solution Methods

An algorithm that calculates a solution of the optimization problem with a required accuracy is a solution method for a class of optimization problems. A significant effort has been made since Dantzig's development of simplex method [23] for linear programming problems (LPPs) to design algorithms for different classes of optimization problems (non-linear), analyze their properties and develop robust software implementations. The efficiency of these algorithms depends on factors such as particular forms of objective and constraint functions, the number of decision variables and constraints, and special problem structure such as sparsity. A problem is sparse if each constraint function depends on only a small number of variables [61].

As mentioned earlier, a significant difficulty in optimization is that it is not possible to solve all problems efficiently using the same method. There are two approaches to address this fundamental issue.

1. If the problem is non-convex then, in most cases, a special optimization method has to be developed to solve this problem efficiently. Good differentiable large scale optimizers now exist in KNITRO that is a solver for non-linear optimization problems [104].

2. An approximation of the original model that is accurate enough for serving the purposes of the application needs to be obtained. It can be solved efficiently using the existing optimization methods. Generally, **linear** and

convex optimization problems can be solved efficiently by a variety of modern optimization techniques [12].

Most problems cannot be accurately formulated as linear or convex problems. Moreover, it is not always straightforward to find a suitable convex approximation to a given optimization problem.

2.3.1 Linear Optimization Problems

Linear programming problems (LPPs) form an important class of optimization problems. In LPPs, the objective function of decision variables x_1, x_2, $\dots$, x_n is linear and can be minimized or maximized subject to linear equality or inequality constraints on x_i, $i = 1, \dots, n$. It is worthwhile noting that LPPs are, of course, convex optimization problems that can usually be stated in a standard form:

$$\text{minimize } (\mathbf{c}^T\mathbf{x}) \quad \text{subject to} \quad A\mathbf{x} = \mathbf{b}\,, \quad \mathbf{x} \geq 0\,, \tag{2.2}$$

where $\mathbf{c} = (c_1, c_2, \dots, c_n) \in \mathbb{R}^n$, $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$, A is an $m \times n$ matrix and $\mathbf{b} = (b_1, b_2, \dots, b_m) \in \mathbb{R}^m$. Basic terminology used in formulating LPPs includes the following definitions:

- ***Feasible Solution***: if $\mathbf{x}$ satisfies $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq 0$ then, $\mathbf{x}$ is feasible.
- ***Feasibility of an LPP***: if there exists a feasible solution then, (2.2) is feasible, otherwise it is infeasible.
- ***Optimal Solution***: x^* is said to be an optimal solution to (2.2) if it is a feasible solution of (2.2) such that $\mathbf{c}^T x^* = \min\{\mathbf{c}^T\mathbf{x} : A\mathbf{x} = \mathbf{b}\,, \mathbf{x} \geq 0\}$.
- ***Unbounded Problem***: if $\forall \lambda \in \mathbb{R}\ \exists$ a feasible x^* : $\mathbf{c}^T x^* \leq \lambda$ then, (2.2) is an unbounded problem.

Note that for LPPs the local optimality implies global optimality. If an LPP has a local solution then, it must have an optimal solution that corresponds to an extreme point. There may be multiple optimal solutions but only one optimal solution value. The value of the objective function for the optimal solution is said to be the value of the linear program. This may occur when the objective function hyperplane is parallel to a binding constraint facet.

There is no simple closed form solution for an LPP but there are a variety of effective methods for solving it, including Dantzig's simplex method [23] and the more recent interior point methods [12].

Theorem 2.3.1 (Fundamental Theorem of Linear Programming)**.** *For an LPP with a feasible domain $\mathcal{D}$ containing at least one extreme point, the optimal objective value is either unbounded or is achievable at one extreme point of $\mathcal{D}$* [21].

The set of linear constraints defines a polyhedron that constitutes the feasible region. A polytope is a geometric object with flat sides, which exists in any general number of dimensions. A polygon is a polytope in two dimensions, a polyhedron in three dimensions and so on in higher dimensions such as a polychoron in four dimensions. Based on Theorem 2.3.1, we can restrict ourselves to the vertices of this polyhedron. The fact that the number of vertices is finite guarantees the termination of any algorithm that explores all vertices.

2.3.1.1 Simplex Method

The simplex method (SM) is one of the most significant algorithmic developments of the 20th century that was introduced in 1947 by George Dantzig [23] as an efficient algorithmic tool for practical decision making. The SM is based on the Fundamental Theorem of Linear Programming (Theorem 2.3.1), which states that the optimal solution, if it exists, is at one of the vertices of the feasible polytope. Thus, it reaches a solution by visiting a sequence of vertices of the polyhedron, moving from each subsequent vertex to an adjacent one characterized by a better objective function value. Since the number of vertices is finite, termination of the procedure is guaranteed. Degeneracy occurs when a vertex in $\mathbb{R}^m$ is defined by $p > m$ constraints, and a step of length zero may be produced. In this case, the SM does not actually move away from the current vertex and thus, no improvement in the objective function value can be achieved. Moreover, given the monotonic method of choosing the next vertex, the set of possible vertices decreases after each iteration in the non-degenerate case.

In terms of practical efficiency, the SM was considered for an extended time as the undisputed method for solving linear programming problems. Klee and Minty [51] were the first to provide an example of pathological behavior of the SM. Their example shows that the SM did not have polynomial complexity of $\mathcal{O}(n^k)$. Therefore in some practical examples, it may be necessary to examine most if not all of the vertices. This can be inefficient given that the number of vertices grows exponentially. Thus, the SM has exponential complexity. Researchers have long tried to develop solution algorithms whose computational complexity times are polynomial functions of the problem size.

2.3.1.2 Interior Point Methods

Interior point methods (IPMs) were developed in the 1960s and the beginning of the 1970s as methods to solve non-linear programming problems with inequality constraints. However, they came out of favor and received less and less attention because of their inefficiency and the presence of strong competitors such as the sequential quadratic programming [97]. In the view of solving LPPs they could not compete with the SM. They were only regarded as theoretical alternatives, not practical substitutes, for the SM.

In 1979 Khachiyan [49] discovered first polynomial alternative to the SM called an ellipsoid algorithm. However, in most practical examples this method was less efficient than the SM. A new polynomial time algorithm based on IPMs for solving LPPs was presented by Karmarkar in 1984 [48]. This method was claimed to be up to 100 times faster than the SM.

The big difference between an IPM and the SM lies in the nature of trial solutions. The SM finds the optimal solution among corner points (vertices) of the feasible region while an IPM converges towards the interior of an optimal facet. The computational complexity is another important difference between these two methods: the SM has exponential time complexity while an IPM has polynomial time complexity [24]. There are classes of problems that are best solved with the SM, and others for which an IPM is preferred. Size, structure and sparsity play the major role in the choice of an algorithm for computations. As a rule of thumb, if the problem size is small, the SM is superior in most cases while for larger problems IPMs are the methods of choice [21]. IPMs can be applied to a wide range of problems. In particular, they have been successfully applied to complementarity problems, quadratic programming and convex non-linear programming [21].

2.3.2 Least Squares Problems

The least squares method has been a standard procedure for the analysis of data from the beginning of 1800s. It was first suggested by Adrien-Marie Legendre [5]. It was used when Gauss predicted the orbit of the asteroid Ceres in 1801 [10]. Nowadays this method is used in various computational applications such as signal and image processing [99]. Another important applications of a least squares problem (LSP) is data fitting.

A linear least squares problem (LLSP) is an optimization problem with no constraints and an objective function that is a sum of squares of terms of the form $a_i^T\mathbf{x} - b_i$, where a_i^T are the rows of M, $\mathbf{b} = (b_1, b_2, \dots, b_m) \in \mathbb{R}^m$ and b_i are the components of $\mathbf{b}$. The aim is for given $M \in \mathbb{R}^{m\times n}$(with $m \geq n$) and $\mathbf{b} \in \mathbb{R}^m$ to find the vector $\mathbf{x} \in \mathbb{R}^n$ such that it minimizes

$$f_0(\mathbf{x}) = ||M\mathbf{x} - \mathbf{b}||_2^2 = \sum_{i=1}^{m}(a_i^T\mathbf{x} - b_i)^2. \tag{2.3}$$

In (2.3), $r = M\mathbf{x} - \mathbf{b}$ is called the residual or error. The least squares method leads to the normal equation

$$(M^TM)\mathbf{x} = M^T\mathbf{b}. \tag{2.4}$$

Its analytical solution is

$$\mathbf{x} = (M^TM)^{-1}M^T\mathbf{b}\,, \tag{2.5}$$

where M is a full-rank matrix. There are reliable and efficient algorithms for solving LSPs with high accuracy. This kind of problems can be typically solved in the complexity time of order (n^2m). Moreover, by exploiting some special structures such as sparsity of the coefficient matrix M, a large LSP can be solved much faster [12].

There are two classes of LSPs: linear least squares problems (LLSPs) and non-linear least squares problems (NLSPs). The core calculations in both classes of problems are similar. Identifying an optimization problem as an LLSP is straightforward by verifying that the objective function is a quadratic function and testing whether the associated quadratic form is semidefinite [12]. Mathematically speaking, an LLSP is the problem of approximately solving an overdetermined system of linear equations and is convex. A system is overdetermined if there are more equations than unknowns. The NLSP is an LSP which is used to fit a set of m observations with a model that is non-linear in n unknown parameters $(m > n)$. Such problems are usually solved by iterative methods so that at each iteration the system is approximated by a linear one. There are various standard techniques to improve the applicability of LSPs.

- **Weighted least squares (WLS)**: in this technique, $f_0(\mathbf{x}) = \sum_{i=1}^{m} w_i(a_i^T\mathbf{x} - b_i)^2$ is minimized. The positive weights $w_1, \ldots, w_m$ are chosen to reflect differing levels of concern about the sizes of the terms $(a_i^T\mathbf{x} - b_i)$.

- **Regularization**: in this technique, a positive multiple of the sum of squares of the variables is added to the cost function (2.3) so that

  $$f_0(\mathbf{x}) = \sum_{i=1}^{m}(a_i^T\mathbf{x} - b_i)^2 + \rho\sum_{i=1}^{n} x_i^2\,,$$

  and the parameter $\rho \geq 0$ is chosen to ensure the desired trade-off between minimizing the original objective function (2.3) and keeping $\sum_{i=1}^{n} x_i^2$ reasonably small [12]. The LSPs introduced here will be studied in later chapters with more technical detail.

- **Least Absolute Shrinkage and Selection Operator (LASSO)**: this technique results in a solution with few nonzero entries in $\mathbf{x}$.

  $$\min \frac{1}{2}||M\mathbf{x} - \mathbf{b}||_2^2 + \zeta||\mathbf{x}||_1\,,$$

  where ζ is a slack variable and $||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$. A slack variable is added to an inequality constraint in order to convert it to an equality one. Moreover, it is a trade off between fitting the data perfectly and employing a sparse solution. This technique is also known as sparse problems.

2.3.3 Convex Optimization Problems

Convex optimization can be defined as a combination of convex analysis and numerical computation. Convex optimization provides additional tools that extend our ability to solve problems such as LSPs and LPPs. It has many applications in signal processing, networks, control, computer science, operation research, statistics, economics, data analysis, modeling and information theory. It has also found a wide range of applications in global optimization and combinatorial optimization where it is used to find bounds on the optimal value as well as approximate solutions. A fundamental property of convex optimization problems is that any locally optimal point is also globally optimal [12]. Elementary definitions used in convex optimization problems are as follows:

- The set $\Omega \subseteq \mathbb{R}^n$ is *convex* if, for all $\mathbf{x}, \mathbf{y} \in \Omega$ and for $\lambda \in [0, 1]$, the vector $\lambda\mathbf{x} + (1 - \lambda)\mathbf{y} \in \Omega$.
- The intersection of any number of convex sets is convex.
- The intersection of all the convex subsets of $\mathbb{R}^n$ containing $\mathbf{C}$ ($\mathbf{C} \subseteq \mathbb{R}^n$) is called the *convex hull* of $\mathbf{C}$.
- If $(\bar{\mathbf{x}}^1, \bar{\mathbf{x}}^2, \ldots, \bar{\mathbf{x}}^k) \in \mathbb{R}^n$, then, a *convex combination* of a set of vectors is a vector of the form

$$\sum_{i=1}^{k} \lambda_i \bar{\mathbf{x}}^i\,, \quad \text{where} \quad \sum_{i=1}^{k} \lambda_i = 1\,, \quad \lambda_i \geq 0.$$

- Convex combinations of a set of vectors belonging to a convex set of $\mathbf{C}$ also belong to $\mathbf{C}$.
- Let $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ be a function. Then, f is said to be a *convex function* if for $0 \leq \lambda \leq 1$, $f[\lambda\bar{\mathbf{x}} + (1 - \lambda)\bar{\mathbf{y}}] \leq \lambda f(\bar{\mathbf{x}}) + (1 - \lambda) f(\bar{\mathbf{y}})$.

In a convex optimization problem, the objective and inequality constraint functions in (2.1) are convex so that functions $f_0, \ldots, f_m$ satisfy the inequality

$$f_i(\alpha\mathbf{x} + \beta\mathbf{y}) \leq \alpha f_i(\mathbf{x}) + \beta f_i(\mathbf{y}), \tag{2.6}$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1,\ \alpha \geq 0,\ \beta \geq 0$, and the equality constraint functions should be affine. Two special classes of the general convex optimization problem are an LPP (2.2) and LLSP (2.3). Although there are reliable and efficient methods for solving convex optimization problems, there are no closed form solutions. This is in contrast to an LLSP whose solution is given analytically by (2.5) assuming M is a full-rank matrix.

2.3.4 Non-linear Optimization Problems

If the objective or any of the constraint functions are non-linear and not known to be convex then, the problem (2.1) is a general non-linear optimization problem. It may have many local minima that might not be the global ones. There are no efficient methods for solving such problems in general. A large number of optimization methods have been developed to solve such problems. Traditional methods to solve them include both local and global optimization methods. Furthermore, the feasible regions of non-linear constraints are difficult to find. Therefore non-linear optimization problems are difficult to solve [77].

Some local optimization methods require differentiability of the objective and constraint functions to solve the problem. These methods need an initial guess for the optimization variables $\mathbf{x} \in \mathbb{R}^n$. Their choice is critical since it can greatly affect the objective value of the local solution. The local optimization algorithms are also sensitive to various algorithm parameter values that provide information about how far from a global solution the local solution is.

Finding a good initial guess, selecting an appropriate algorithm and adjusting algorithm parameters are all issues for local optimization methods therefore solving a non-convex problem by a local optimization method is not as straightforward as solving an LSP, an LPP or a convex optimization problem. Formulating a practical problem as a non-linear optimization problem is relatively straightforward because the differentiability of the objective and constraint functions is the only requirement for most of local optimization methods. Many of these problems are modeled as non-convex and have multiple optima. While local optima can be found using traditional local optimization methods, for instance, the gradient descent method, the conjugate gradient method or the Newton's search, attempts to use these methods for global optimization typically fail. To address this issue, global optimization methods have been developed that are able to escape local optima. However, these methods generally do not perform well on a large scale problem.

Generally, the findings of this investigation demonstrate that

- a problem can be formulated easily as a non-convex problem but attempts to solve it through local optimization methods are challenging tasks;
- the problem formulation in convex optimization is a challenging task but there exist many reliable and polynomial time algorithms to globally solve it [12].

Therefore these facts motivate us to pursue further studies in convex optimization. It can also be employed in problems that are not convex to provide suboptimal solutions [12]. Furthermore, many interesting and important problems in biomedical signal processing can be posed as convex optimization problems.

2.4 Biomedical Signal Processing

The use of optimization methods is ubiquitous in signal processing that is an area of applied mathematics dealing with signals in either discrete or continuous form. In signal processing, an algorithm is used to extract some desired signal or information from a received noisy signal. Signal processing design and analysis frequently begin with the statement of a criterion of optimality followed by an optimization procedure.

The application of optimization methods in signal processing has expanded greatly in recent years and therefore we will not try to cover the field completely but only focus on biomedical signal processing. Biomedical signals are used by physicians for monitoring health of patients. They originate from various biological systems that include the digestive, circulatory, respiratory, musculoskeletal and nervous systems [62]. These signals are recorded for various physical measurements e.g. temperature, pressure or electric potential [19].

Bioelectrical signals are specific types of biomedical signals and are obtained by electrodes that record the variation in electrical potential generated by a physiological system. Electroencephalogram (EEG), electrooculogram (EOG), electrocardiogram (ECG), electrogastrogram (EGG) and electromyogram (EMG) are a few bioelectric signals which are used regularly by medical practitioners. EEG is used to monitor neural activity, EOG to record eye movement, ECG to assess heart functioning, EGG to measure stomach wall nerve activity and EMG to measure muscular activity.

A goal of biomedical signal processing is to develop the procedure for automatic detection and classification of pathological events taking the advantage of digital signal processing and modern powerful digital computers.

Electrical signals created by brain depict not only the brain function but also the status of whole body. This belief provides the motivation to apply convex optimization as signal processing methods to EEG signals measured from the brain of a human subject. In the following section, background information of EEG is provided in detail.

2.5 Electroencephalogram (EEG)

The human brain controls all functions of the body such as thinking, learning, remembering, speaking and decision making. It consists of about 100 billion nerve cells called neurons. Neurons create very small electrical signals that form patterns called brain waves. They transfer information to and from the brain and transmit electrical potentials to other cells along axons. Then, the human brain structure and functions are indeed complex.

EEG is an electrical activity of the human brain that can be recorded graphically and is used as an important tool for studying neural activities. The electrical activity is generated by firing of neurones of the human brain due to internal or external stimuli to control different bodily actions. EEG is widely used by physicians to study and analyze neural activities recorded with electrodes placed on scalp, directly on the cortex. It enables clinician and physicians to diagnose neurological disorders such as epilepsy, sleep disorder, brain tumor, head injury, and so on.

In 1924, Hans Berger, a German neurologist, first introduced an EEG machine to the world. It employed to amplify the electrical activity measured on human scalp [20].

He announced that the brain currents changed according to the functional status of the brain for instance in sleep, epilepsy, lack of oxygen and anesthesia.

To record brain activities a number of electrodes are placed to different locations on the surface of the scalp such that per pair of electrodes is connected to an amplifier and an EEG machine. So, the electrodes detect brain waves and the EEG machine amplifies the signals and records them in a wave pattern on a graph paper or a computer screen. A channel is made by a pair of electrodes. It creates a signal during an EEG recording. Brain patterns that form wave shapes are usually sinusoidal. The amplitude of an EEG signal normally range from 1 to $100\,\mu$V in a normal adult [78]. There are five brain waves that are identified by the frequency and the amplitude.

- **Delta** wave is within the range of $0.5 - 4$ Hz. It has a large amplitude recorded by scalp electrodes and emerges during deep sleep. This wave indicates serious brain disorders or cerebral damage [74, 78].
- **Theta** wave is within the range of $4 - 7.5$ Hz. It happens during somnolence. Although it occurs during sleep, it is recorded for some purposes from individuals who are awake. For instance, there are cases when theta wave appears during emotional responses to frustrating situations [74, 78].
- **Alpha** wave is those frequencies within the range of $8 - 13$ Hz. It appears in the posterior half of the head when an adult is awake but relaxed with eyes closed. It is usually arisen from stress and tension [74, 78].
- **Beta** wave lies within $14 - 26$ HZ and tends to have a low amplitude. It is generated when the brain is engaged in mental activities such as remembering or retrieving memories. Therefore it appears when the cortex is aroused to a higher state of alertness [74, 78].
- **Gamma** wave or fast beta wave consists the frequencies above 30 Hz (mainly up to 45 Hz) and has a very low amplitude. Despite the fact that it happens rarely, its detection can be used for testimony of certain brain diseases. It is associated with the state of active information processing of the cortex [35, 74, 78].

Artifacts are electrical signals in an EEG that are not originated from the cerebral origin. They arise from many sources such as the heart, eye movement, and muscles and may affect the outcome of the EEG recordings. The problem of distinguishing between the genuine EEG activity and artifacts are arisen during the analysis of EEGs. Their identification and elimination from EEG signals are difficult tasks and need EEG experts [35, 78].

The literature has emphasized the significance of the EEG signal analysis in order to diagnose neurological disorders such as epilepsy, sleep disorder, comma, brain tumor and head injury. In this thesis, we will not try to cover the field completely but only focus on sleep disorder and epilepsy as applications of EEG signals. More details are provided in the following subsections.

2.5.1 K-complex Detection for Sleep Staging of EEG

Currently, there is an alarming number of people in the world suffering from sleep disorders. The diagnosis of such disorders is performed in a test called polysomnogram (PSG). PSG studies a series of biomedical signals such as brain activity (EEG), muscle movements (EMG), heart beat (ECG) and eye movement (EOG) for diagnosis of sleep disorders. The analysis of an EEG, in particular, the sleep stages identification (scoring), is an active research area in the biomedical signal processing [100, 101].

The usual method for sleep stage identification is visual (manual) inspection of an EEG signal by a medical doctor. The sleep stage scoring is based on the standardized scoring Rechtschafen and Kales (R&K) rules [45]. One of the major deficiencies of these rules is arbitrarily defined thresholds for sleep stage identification. This can lead to unreliable results and poor agreement between scorers.

Recently, the American Academy of Sleep Medicine (AASM) [41] developed an updated set of rules to identify sleep stages. There are two main purposes for these changes. First of all, the new rules are simpler and easier to apply; and therefore a higher level of agreement between different manual scorers can be achieved. Second, the updated rules are more straightforward to automate.

Human (adult) sleep consists of two main parts: Rapid Eye Movement (REM) sleep and Non-REM (NREM) sleep. Typically, patients begin the sleep cycle with a period of NREM sleep followed by a very short period of REM sleep. The AASM divides NREM into three further stages [41].

- NREM stage 1 is the transition from waking to sleep.
- NREM stage 2 is signaled by K-complexes in the EEG.
- NREM stage 3 is called Slow-Wave-Sleep (SWS), deep sleep or delta sleep.

After NREM stage 3, the patient returns to NREM stage 2 and then, enters REM stage.

The AASM rules [41] defined the K-complex as a *"well-delineated negative sharp wave immediately followed by a positive component standing out from the background EEG, with total duration* ≥ 0.5 *second, usually maximal in amplitude when recorded using frontal derivations"*. One of the main characteristics of a K-complex is a sudden increase in the signal amplitude. Reliable detection of K-complexes in an EEG signal is essential for sleep stage scoring since they constitute one of the main markers of the transition from NREM stage 1 to NREM stage 2. Unfortunately, their visual (manual) identification is very time consuming and rather subjective (there are typically 1 to 3 K-complexes per minute in NREM stage 2 of young adults [52]).

K-complexes consist of an initial small negative, somewhat sharp wave, followed by a large positive wave. This definition has been established by medical practitioners in [45]. Informally, this means that the amplitude increases abruptly and then, returns back to the original value. Since EEG signals are non-linear, non-stationary and not repeatable, K-complexes have a wide variety of shapes and are difficult to distinguish from other EEG waves. There have been several attempts to address this issue.

In 1970, Bremer *et al.* [13] worked on the detection of K-complexes that were related to the study of arousal during sleep. They were the first to propose an electronic system using filters, pulsers, threshold detectors, and both analog and digital logic elements. They obtained a classification accuracy of 68% for considering sleep stage 2.

Jansen [42] tried to perform a direct automatic recognition of K-complexes by using artificial neural networks (ANNs) where the inputs were the samples of the band-filtered EEG. However, he achieved poor performance with the detection rate ranging from 42% to 67%.

Da Rosa *et al.* [73] proposed a detector based on a K-complex model. A true positive (TP) second was counted when the method output indicated a K-complex simultaneously with a scored K-complex. The true positive rate (TPR) was the number of detected K-complex seconds divided by the number of scored K-complex seconds and the false positive rate (FPR) was the number of false positive findings divided by the number of all findings by the method. If the method output indicates a K-complex and there is no K-complex scoring, a false positive (FP) finding is counted. The results of true positive and false positive rates were respectively 89% and 49%.

Another attempt to identify K-complexes was made by Bankman *et al.* in 1992 [6]. They extracted 14 features of a signal from the EEG using a visual analysis and then, fed them into a neural network. They reported 90% true positive and 8% false positive rates.

Henry *et al.* [40] studied K-complexes in 1994. They showed that the performance of wavelet was superior as far as the low missed detection and the low false alarm were concerned. However, in the K-complex detection in sleep EEG signals, their methods were unable to distinguish events when the patterns were close.

Tong and Ishii [87] reported a K-complex detection algorithm using discrete wavelet transform. They obtained 87% true positive and 10% false positive rates. They assumed that the K-complex is always overridden by a spindle whereas Jansen and Desai [43] stated that there was no fixed relationship between these two events. Sleep spindles are high frequency ($12-14$ Hz), relatively low amplitude bursts of sinus like waves with minimum duration of $500\ msec$ that are present in sleep stage 2.

While many studies were carried out to automate sleep staging, no standard algorithm has been accepted by the medical community. At the same time the visual (manual) scoring of K-complex is an expensive and time consuming procedure and the reported agreement between different human expert scorers is poor. According to the literature, medical doctors still report that the accuracy of identification is not satisfactory as the process remains subjective [2, 41]. Therefore an accurate method for automatic detection of K-complexes would be advantageous. This automation would reduce the number of manual tasks significantly, thereby making the process more reliable and cost efficient [101].

A few further algorithms dealing with the detection of K-complexes have been recently proposed and reported. A non-smooth optimization approach was applied for automatic detection of K-complexes by Moloney *et al.* [57] in 2011. A non-smooth optimization problem refers to the more general problem of minimizing functions that are typically not differentiable at their minimizers. Their method is based on minimizing the deviation between the actual EEG curve and the wave shapes (model patterns). It was used to extract the wave features and thus, enables one to significantly reduce the dimension of the problem. The classification algorithms implemented in WEKA [95] were used over the obtained set of features. WEKA is a collection of machine learning algorithms for data mining tasks [95]. The EEG curve was modeled as the sum of two sinusoidal curves. The amplitude of each curve was approximated as a piece-wise linear function. The frequency and the phase shift were modeled as additional variables and were not constants. Although the numerical experiments of their work showed the robustness of this procedure, the proposed algorithm was time consuming [57].

2.5.2 Epileptic Seizure Detection in EEG Signals

Epilepsy is a neurological disorder disease that manifests in about one percentage of the world's population [75]. It is characterized by a recurrent seizure that happens when the neurons generate abnormal electrical discharges from brain cells. A seizure is experienced in about 5% of individuals in their life [59] and approximately 30% of patients have disobedient seizure that can lead to the neural tissues disorders [103]. The seizure can be treated by medication in 70% of patients [29].

The seizure can cause physical changes in behavior and movements, loss of consciousness, muscle spasms, strange emotions and even death. Therefore detection of epilepsy is still a challenging issue for medical diagnosis of epilepsy. An EEG is a well known tool for identification of epileptic seizure since it measures the voltage fluctuations of the brain [27, 86] and provides important information about epileptic activities. Visual detection of epileptic seizure in an EEG signal is being time consuming and causing fatigue and requires highly trained practitioners. Different steps like preprocessing, features extraction and classification can be involved in an epileptic seizure detection technique.

There have been various attempts for automatic detection of epilepsy based on Wavelet Transforms [1, 50, 98], Artificial Neural Networks [16, 80] and Genetic Programmings [8].

Panda *et al.*, 2010 [68] applied discrete wavelet transform and a classifier called support vector machine (SVM) to compute various features like energy, entropy and standard deviation. They obtained the classification accuracy of 91.2%.

The classification accuracy of 96.7% was obtained through the mixed-band wavelet chaos neural network method by Dastidar *et al.*, 2007 [31]. They used wavelet transformation to break up the EEG signals into different range of frequencies and three features, namely standard deviation, correlation dimension and the largest Lyapunov exponent, were used and different methods employed for the classification.

To decompose the normal and epileptic EEG epochs to various frequency bands and to find optimal features subsets which maximize the classification performance, fourth level wavelet packet decomposition method was proposed by Ocak [65]. The classification accuracy of this method was 98%.

Polat and Gne (2007) [69] applied two stage processes. First one was Fast Fourier Transform (FFT) as a feature extractor and second one was the decision making classifier. They got a classification accuracy of 98.72%.

Bhardwaj *et al.*, 2015 [8] applied an automated detection of epileptic seizures in EEG signals using Empirical Mode Decomposition (EMD) for feature extraction and proposed a Constructive Genetic Programming (CGP) approach for classifying the EEG signals. The classification accuracies of 100% and 99.41% (an average classification accuracy) were obtained from one Genetic Programming (GP) run and 100 GP runs, respectively, through a CGP for 10-fold cross validation scheme.

A new method for classification of ictal and seizure-free EEG signals was presented by Pachori and Patidar in 2014 [67]. The proposed method was based on the EMD and Second-order Difference Plot (SODP) of Intrinsic Mode Functions (IMFs). They computed the 95% confidence ellipse area parameters for ictal and seizure-free classes using SODP of IMFs for various window sizes. The best average classification accuracy of 97.75% was obtained for IMF1 and IMF2 with window size of 1000. The maximum classification accuracy was 100%.

To the best of our knowledge, the best method in terms of classification accuracy was developed by Bajaj *et al.*, 2012 [4]. For the classification of seizure and non-seizure EEG signals they applied least square SVM (LS-SVM) and they got the classification accuracies of 98% to 99.5% using radial basis function (RBF) kernel and 99.5% to 100% using Morlet kernel.

Multiple researchers have been using the classification accuracy as the main criterion characterizing their performance. The evidence from this study shows that this criterion is based on the proportion of correctly and incorrectly classified segments. It is worthwhile mentioning that the above techniques used the Bonn University dataset [3]. High classification accuracies may be acquired owing to the existence of unbalanced datasets where a disproportionately large amount of segments (instances) belongs to a certain class although the proposed classifier may not necessarily be good. As an alternative to classification accuracy, the area under the Receiver Operating Characteristic (ROC) curve can used to assess the classification accuracy where there exist unbalanced datasets [102]. The ROC curve is produced by plotting the fraction of true positive rate (TPR) against the fraction of false positive rate (FPR) as the threshold for discrimination between two classes is varied [102]. The definitions of TPR and FPR are provided in Section 3.9 of Chapter 3.

More research should be conducted to detect epileptic seizures by analyzing EEG signals since it is a challenging task due to inconsistency of signals in patient's sex, seizure's type, patient's age and so on. It should be noted that providing a good trade off between a high classification accuracy and a low false positive rate (FPR) is a difficult problem in classification purposes.

To address these issues, we propose a novel method based on convex optimization approaches and different classifiers from WEKA [94] for automatic detection of K-complexes and epileptic seizures in EEG signals. This work is aimed at discovering the optimal approximation of an EEG signal under a sinusoidal modeling function and consistency improving the performance of convex optimization methods in classification problems compared with the original signal.

2.5.3 Feature Extraction and Classification of EEG Signals

In general, EEG signals are recorded with a number of electrodes ranging from 1 to 256 and with a sampling frequency ranging from 100 Hz to 1000 Hz. Therefore recording a brain activity results in obtaining a very large set of data involving a large number of variables.

The analysis of complex data obtained from an EEG signal requires a long computational time and a large amount of memory. Moreover, a classification algorithm may overfit the training segments and have a poor classification and predictive performance. Overfitting happens when a model has too many variables to the number of observations.

Challenging this issue, one needs to reduce the size of complex data containing features of original signal whereas it is still describing the properties of signal with sufficient accuracy. This procedure is called "feature extraction". It contains the related information from the data and reduces the amount of resources needed to describe a signal. The features can be frequency, phase shift, amplitude and so on.

If the extracted features from a signal are not accurate enough to describe the original signal, the classification algorithms will not recognize those features appropriately. So, in order to enhance the performance of classification algorithms (classifiers), selecting and extracting good features are essential. According to the literature, picking out a good feature extraction (preprocessing) method is more important than selecting a good classification algorithm (classifiers) [55].

To this end, algorithms that can extract the relevant information from an EEG signal should be designed as feature extraction methods to improve the performance of classifiers and to facilitate the consequent implementation of the classifiers. Although there exist many methods to extract the essential features of an EEG signal, identification of the most efficient ones is a difficult task because of a shortage of comparisons.

The classification process takes place into two stages called feature extraction and classification. In the first stage, the essential features are extracted that are needed to describe a signal accurately. In the second one, a classifier determines to which class the segments of an EEG signal belong based on the extracted features obtained from previous stage.

There exist two categories of classification called supervised and unsupervised classifications. In the supervised classification, recordings of an EEG signal are assigned to the class labels while in the unsupervised one, recordings of an EEG signal (observations) are not assigned to a known class [78].

In the supervised classification, the information regarding to the class label is given within the recordings of an EEG signal. Each dataset is partitioned into training and test sets. A classifier is constructed through the training set. To evaluate the performance of the classifier, the test set is used. The training set in the supervised classification contains a set of segments that have class labels [14, 28].

To express the above procedure mathematically, let

$$Tr = \{(y_1, c_1), (y_2, c_2), \ldots, (y_N, c_N)\},$$

be a given training set that contains $y_1, y_2, \ldots, y_N$ and $c_1, c_2, \ldots, c_N$ as observations (EEG recordings) and class labels of the observations respectively. For instance, if the problem is K-complex detection then, y_i are EEG recordings and c_i is either K-complex or non-K-complex for $i = 1, 2, \ldots, N$. Assume that the observations (EEG recordings) and class labels are presented as real values and numbers respectively then, $y_i \in Y$ (real valued observations) and $c_i \in C$ (real numbers). The goal of the supervised classification is to determine the transformation between the feature space Y and the class label space C such that $f : Y \longrightarrow C$. If there exist a finite number of elements in the class space then, there is a classification task problem.

In the binary classification problem, the classes are divided into two categories called non-target and target classes. Therefore these classes are represented as $C = \{-1, +1\}$ where -1 represents the non-target class. When a classifier is trained over the training set it is ready to assign class labels to the features on the test set with unknown class labels. The aim of this learning procedure is to maximize the accuracy on a test set. In this research, the supervised procedure is used to classify an EEG signal in presence of transient events.

In the unsupervised classification, there is no information about the class labels within the dataset. In this kind of classification the dataset is grouped into classes according to the measure of inherent capability such as the distance between instances. The training set has not been labeled therefore the class labels on the test set can be determined by finding the inherent properties of the dataset [14, 28].

2.6 Summary

In summary, a major relevant literature of optimization methods in the biomedical signal processing has been reviewed. A background about EEG signals has been provided. The feature extraction and classification of EEG signals are also discussed. Two important problems in this field that are K-complexes and seizures detection have been considered and studied. The literature from classification of an EEG signal indicates that there are drawbacks and limitations associated with the existing methods for instance, long computational time, false classification of a severe noise or a sleep spindle as a K-complex, the existence of unbalanced datasets and the lack of investigations in performances consistency of corresponding methods for the seizure detection. Challenging these issues, development of convex optimization models to extract the essential features of an EEG signal is required in order to

- detect the transient events (K-complexes and seizures) automatically;
- approximate an EEG signal accurately;
- improve the classification accuracy and reduce the dimensionality of the dataset.

The developed models can help to diagnose of sleep disorders and the epileptic seizures disease in clinics. In Chapter 3, a novel method based on the sequence of linear least squares problems and the uniform (Chebyshev) approximations is introduced to extract the essential features of an EEG signal in presence of K-complexes and seizures.

Chapter 3

Characterizing an EEG Signal Using Convex Optimization Modeling

3.1 Introduction

This section discusses the methodology of the research and presents the direction of the study. The development of feature extraction methods is essential for the analysis of EEG signals in order to improve the accuracy in classification problems. These methods enable one to extract the essential features of an EEG signal that are crucial for diagnosis and evaluation of brain diseases.

The analysis of complex data captured from EEG recordings requires the long computational time. Therefore the utilization of developed methods significantly reduces the dimension of the classification problem and avoids the overfitting of a classier. The main purposes of the research are to

- find the optimal approximation of an EEG signal under a sinusoidal modeling function that describes the original signal accurately and
- improve the performance of proposed methods in classification problems compared with the original signal.

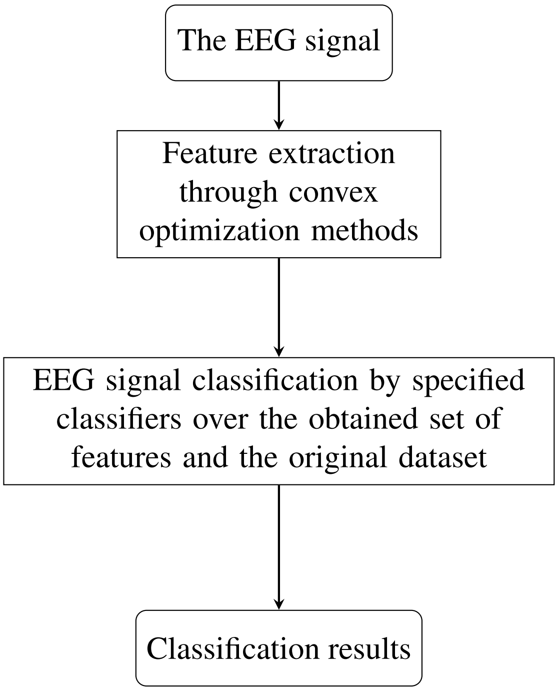
The EEG signal
Feature extraction through convex optimization methods
EEG signal classification by specified classifiers over the obtained set of features and the original dataset
Classification results

Figure 3.1: Methodology Framework

First, four convex optimization models are developed to extract the key features of EEG signals. Second, twelve classification algorithms used in [57] are employed over the obtained set of features and the original dataset to evaluate the performance of proposed methods with respect to classification accuracy of an EEG signal. Results for the research will be collected through performing the proposed methods over two different types of datasets namely EEG K-complexes and epileptic seizures datasets. The general steps of the proposed methods for classification of an EEG signal in presence of transient events (K-complexes and seizures) is depicted in Figure 3.1.

3.2 Motivation

To date, a few techniques have been proposed to extract the essential features of EEG signals dealing with detection of K-complexes recently.

Sukhorukova *et al*. [58] in 2010 applied non-smooth optimization techniques as feature extraction methods to identify sleep stages automatically. Their methods is based on minimizing the deviation between the original and approximated signals. The EEG curves were modeled as the sum of two sine curves. The signal amplitude was modeled as a piecewise linear function. In the classification stage, they applied an ANN over a set of extracted features obtained in the feature extraction stage. They used Short-Time Fourier Transform (STFT) to analyze the frequency domain and refine classification results for specific stages [58].

Recently, Moloney *et al.* [57] in 2011 applied non-smooth optimization to extract the essential features of an EEG signal in order to detect K-complexes. Their method is based on minimizing the sum of absolute deviations between the original EEG signal and the approximated one. The approximated signal was modeled as the sum of two sinusoidal curves and the frequency was approximated through solving a non-smooth optimization problem. The signal amplitude was approximated by a piecewise linear function. The specific classification algorithms were applied over the original dataset and a set of extracted features obtained through non-smooth optimization. The reported numerical results demonstrated that the non-smooth optimization-based preprocessing approach performed well. However, this approach was time consuming due to the non-convexity of problem reformulation and the large number of decision variables to be optimized. It should be noted that a non-convex modulation of a problem can result in multiple optimal solutions that might not be the global ones.

The main difference between the proposed methods in the aforementioned attempts is the frequency modulation. In [58], the STFT was applied to analyze the frequency domain while in [57], the frequency was considered as an additional variable in optimization problem and was optimized through solving non-smooth optimization problems.

A K-complex detection method was applied using a hybrid-synergic machine learning method by Quan Vu *et al.* [93] in 2012. A 10 seconds EEG segment containing 41 subsegments were studied to detect K-complexes. In the feature extraction stage, they divided each subsegment into three intervals. They calculated the difference between the maximal amplitudes along with the positive and negative directions for each interval. They defined 29 visual features heuristically. Those features were planned to capture the visual characteristics of K-complex waveforms. In the classification stage, they developed a Representation Instance-based Classification (RIC) algorithm. The proposed method could identify the EEG subsegments that are most likely to contain a K-complex. However, it did not consider the variability of the observations where the time variability was an essential consideration for describing the characteristics of the original dataset. Although the development of the classification algorithm called RIC was well studied, their feature extraction method requires more consideration. To enhance the performance of the extracted features to automate K-complex detection, the authors suggested to select the features from the designed 29 visual features.

In spite of all the aforementioned advances in the feature extraction stage of classification problems, the proposed methods have following drawbacks to reach the intended results.

- Non-smooth optimization modelings have the long computation time, due to the non-convexity reformulation and the large number of decision variables to be optimized.
- The frequency modulations in the approximated signals described in [57] and [58] formed the optimization problems that were complex and non-convex.
- The time variability of observations was not considered in [93]. This issue significantly affects the efficiency of feature extraction method.

Challenging these issues, a convex optimization-based method as a feature extractor with consideration of the time variability is developed that performs faster and more efficient than the proposed methods in [57] and [58]. Because of the time variability of observations there will be different frequency components at different time. In order to find the range of possible frequencies, different time-frequency testings will be selected based on the specifications of an EEG signal. It is worth to note that any locally optimal solution is also globally optimal in convex optimization problems.

3.3 EEG Signal Modeling

Typically, EEG signals are modeled as a sine wave $A(t)\sin[\omega(t)\,t + \tau(t)]$, where $A(t)$ is the signal amplitude, $\omega(t)$ is the frequency and $\tau(t)$ is the phase shift that does not affect the shape of the wave. The choice of these three functions is important. Consider

$$\min_{A,\omega,\tau} \sum_{i=1}^{N} \{y_i - A(t_i)\sin[\omega(t_i)\,t + \tau(t_i)]\}^2\,, \tag{3.1}$$

where y_i, $i = 1, \ldots, N$ are the EEG recordings at time t_i, with N the total number of recordings. For modeling any kind of signal it is reasonable to consider all $A(t)$, $\omega(t)$ and $\tau(t)$ as functions of t; however, the corresponding optimization problems may become non-convex and have many local minima.

EEG signal specifications indicate that it is a time-varying signal. Therefore there exist different frequency components at different time and the signal's frequencies are time-independent. Because of this uncertainty it is difficult to obtain higher resolution in both time and frequency domains simultaneously and the incident of EEG abnormalities is represented with sudden changes in brain waves. Then, visually frequency identification by screening an EEG signal can not be precise.

The optimization problem (3.1) is generally difficult and time consuming to solve. Although the estimation of the signal amplitude and frequency of an EEG signal is a fundamental task in feature extraction modelings, finding accurate approximations of them are really difficult. To address this issue, one might need to model (approximate) the frequency $\omega(t)$. In this case, the optimization problem (3.1) becomes non-convex. Therefore the explicit optimization of frequencies is very complex problem that also known as Mandelshtam's problem [18, 26]. There have been attempts to approximate the frequency such as the methods presented in [57]. The authors approximated the frequency through solving a non-smooth optimization problem. The main drawback of their problem was long computational time since mathematically, such approximation of frequencies is difficult to model. As a matter of fact, such precision of frequency modulations is rarely justifiable in this type of application.

Medical doctors (practitioners) have specific rules to identify the range of frequencies to detect the feature waves of EEG signals. They do not work with the fractional frequencies. Since our goal is to mimic doctors' decisions, it is enough to choose the most appropriate integer frequency values taken from a carefully specified range. This range is defined by a particular application. Therefore different time-frequency testings can be selected based on the specifications of the proposed EEG signal. This leads to find the range of possible frequencies for the proposed EEG recordings.

An alternative to the non-convex reformulation is to avoid non-convexity from the beginning by modeling the signal amplitude $A(t)$. Hence, our goal is to find a suitable equilibrium, where the corresponding approximations are:

- precise enough to describe the signal and produce high accuracy classification;
- simple enough to be solved accurately and inexpensively on a personal computer used in a medical lab.

3.4 Signal Amplitude Approximation

Capturing extended amplitude changes is a promising approach to detect some transient events in EEG signals. To approximate the amplitude of an EEG signal one possibility is through polynomial functions and another possibility is to use continuous piecewise polynomial ones (also known as splines). The polynomial functions may not be as flexible as spline ones to detect abrupt changes in amplitude, but they are simpler than splines.

Splines require continuity of the function and its derivatives up to a certain order over the complete range while polynomials are used only within an individual interval and do not preserve smoothness at the boundaries between the intervals. The main difference between spline and polynomial approximations is that the former is global while the latter is local in nature. Therefore splines are also polynomials when the number of subintervals is one [63]. In the next two sections, polynomial and spline functions are detailed.

3.4.1 Polynomial Function

Polynomials are a class of functions that are commonly used in calculus and numerical analysis to approximate more complex functions. They are defined by their coefficients (parameters). There are many ways to construct polynomials. The most common way is through monomial basis.

$$P(\mathbf{x}\,,t) = x_0 + \sum_{j=1}^{m_2} x_j t^j\,, \tag{3.2}$$

where m_2 is the degree of a polynomial and $\mathbf{x} = (x_0, x_1, \ldots, x_{m_2}) \in \mathbb{R}^{m_2+1}$ are the polynomial parameters. Polynomials are useful tools for modeling the amplitude due to the fact that they

- are simple;
- have reasonable flexibility of shapes;
- can be differentiated and integrated easily.

Due to these properties the construction of polynomial approximations are computationally inexpensive and accurate.

Theorem 3.4.1. Weierstrass Approximation Theorem [44]. *Assume that f is a continuous function on a closed and bounded interval $[a, b]$. For any given $\varepsilon > 0$, there is a polynomial P with sufficiently high degree m_2 such that*

$$|f(x) - P(x)| < \varepsilon\,. \tag{3.3}$$

This theorem states that every continuous real-valued function defined on a closed and bounded interval $[a, b]$ can be uniformly approximated as closely as desired by a polynomial function however, the degree m_2 may be very high. It is not practical to increase the degree of polynomials since the number of polynomial parameters to be estimated will be high that can leads to a highly unstable model. There exist a number of tools to overcome this difficulty. One of them is through polynomial spline approximations that are introduced in Section 3.4.2.

3.4.2 Spline Function

Polynomials can be pieced together to form spline curves that can approximate any function to any desired accuracy. Splines are suitable candidates to describe abrupt changes in the amplitude therefore it is very natural to use them in amplitude approximating [100, 101]. They describe a polynomial-like behavior of a signal naturally. They are a powerful tool for signal amplitude approximation and have several advantages to other traditional signal processing techniques, including Fourier analysis. Most traditional signal processing approaches are based on smooth functions since optimizing smooth functions is easier than non-smooth ones. Splines are non-smooth functions and therefore they are more suitable to approximate the amplitude [101].

There are many ways to construct splines. One possibility is through a truncated power function [63].

$$S(\mathbf{x}\,,\boldsymbol{\theta}\,,t) = x_0 + \sum_{j=1}^{m_1} x_{1j}t^j + \sum_{l=2}^{n}\sum_{j=1}^{m_1} x_{lj}(t-\theta_{l-1})_+^j\,, \tag{3.4}$$

where m_1 is the degree of a spline, n is the number of subintervals in a D seconds duration of an EEG, N is the number of recordings in a time segment $t = [t_1, t_N]$ when the original signal is recorded, $\mathbf{x} = (x_0, x_{11}, \ldots, x_{m_1 n}) \in \mathbb{R}^{m_1 n+1}$ are the spline parameters, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n-1})$ are the spline knots and

$$(t-\theta_{l-1})_+ = \max\{0, (t-\theta_{l-1})\} = \begin{cases} t-\theta_{l-1}\,, & \text{if} \quad t > \theta_{l-1}\,, \\ 0\,, & \text{if} \quad t \le \theta_{l-1}\,, \end{cases} \tag{3.5}$$

is the truncated power function. Note that in Equation (3.4), any polynomial spline is a piecewise polynomial that may be even discontinuous at its knots [63].

The spline knots can be free or fixed. If the knots are free then, they are considered as additional variables in the optimization problem. Thus, $\mathbf{x} = (x_0, x_{11}, \ldots, x_{m_1 n}, \boldsymbol{\theta})$ and the corresponding optimization problem becomes more complex and non-convex.

The splines are more desirable if the locations of their knots are optimized. Challenging this issue, one might need to work with free knots instead of fixed ones. In this case, the problem becomes non-convex that is complex, computationally expensive and non-smooth.

An alternative to free knots reformulation of S defined in (3.4) is to approximate a signal amplitude as a spline function S with higher dimension fixed knots (more subintervals n) [100, 101]. There are some strategies for the optimal knots localization [70, 72] for those who do not like this simplifying [96]. The splines are detailed in Chapter 4.

3.5 Convex Optimization Models

Feature extraction techniques play an important role in identifying the main characteristics of a signal (extract information from data), reducing the dimensionality of data and classification of a signal. In this section, four convex optimization models as feature extraction methods are developed. The first two models are aimed at minimizing the sum of the squares of the deviation between the original signal y_i and the approximated one. However, this approach leads to the necessity of solving a sequence of linear least squares problems that is a subclass of convex optimization problems. The ultimate goal of the last two ones is to minimize the maximum of the absolute deviation between the original signal and the approximated wave. This leads to solve a sequence of uniform approximation problems that are convex and non-smooth optimization ones.

The signal amplitude is approximated by a polynomial spline S defined in (3.4) whose knots are fixed (equidistant) and a polynomial P defined in (3.2). Assume that the frequency ω and the phase shift τ values form a fine grid. In this case, for each combination of ω and τ values the only function needs to be approximated in (3.1) is the amplitude $A(t)$. Therefore the brain signal (EEG) is modeled as sinusoidal waves and its amplitude is approximated as

$$A(t) = \begin{cases} S(\mathbf{x}\,,\boldsymbol{\theta}\,,t)\,, & \text{and} \\ P(\mathbf{x}\,,t)\,. \end{cases} \tag{3.6}$$

These enable one to

- develop an accurate model for the wave shapes,
- evaluate the performance of simple models in compared with complex ones,

- extract key features of the waves that are crucial for detecting transient events in an EEG signal.

These features are extracted using the following convex optimization models.

3.5.1 Linear Least Squares Optimization Model 1 (LLSOM1)

The EEG signal is modeled as a sine wave

$$W_1 = A(t)\sin(\omega t + \tau)\,, \tag{3.7}$$

where

$$A(t) = \begin{cases} S(\mathbf{x}_1\,, \boldsymbol{\theta}\,, t) = x_0 + \sum_{j=1}^{m_1} x_{1j} t^j + \sum_{l=2}^{n} \sum_{j=1}^{m_1} x_{lj} (t - \theta_{l-1})_+^j\,, \\ \hfill P(\mathbf{x}_1\,, t) = x_0 + \sum_{j=1}^{m_2} x_j t^j\,. \end{cases} \tag{3.8}$$

Then,

$$\mathbf{x} = \begin{cases} \mathbf{x}_1 \in \mathbb{R}^{m_1 n+1}\,, & \text{if} \quad A(t) = S(\mathbf{x}_1\,, \boldsymbol{\theta}\,, t)\,, \\ \mathbf{x}_1 \in \mathbb{R}^{m_2+1}\,, & \text{if} \quad A(t) = P(\mathbf{x}_1\,, t)\,. \end{cases} \tag{3.9}$$

S is the spline function defined in (3.4) whose $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n-1})$ are fixed (equidistant) and P is the polynomial function defined in (3.2) therefore for each combination of ω and τ the corresponding optimization problem is

$$\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - A(t_i)\sin(\omega t_i + \tau))^2. \tag{3.10}$$

This is a linear least squares problem (LLSP) therefore the solution of the original optimization problem (3.1) can be reduced to solve a sequence of LLSPs when ω and τ are constants. We keep the best combination of ω and τ values on a fine grid. This combination gives the minimal value of the objective function. This means that the corresponding parameters (frequency and phase shift) are not optimized formally and a natural question here is whether such kind of approximation is reasonable. To address this question, we have considered these parameters separately in details as follows.

- Phase shift is a parameter that does not affect the shape of approximations (only their locations in time). Therefore even though this parameter is not evaluated very precisely, its influence on the shape of the wave is minimal.
- Frequency contributes significantly to the shape of the corresponding waves. Therefore we need to justify why our grid approximation is reasonable. In this particular application, our goal is to mimic manual rules provided by medical doctors to identify the range of the frequency. Therefore the value for frequency is naturally bounded from above and restricted to integer due to human scorers' perception limitations. Hence, even though we only evaluate frequencies on a grid (rather than exact computation), this accuracy is reasonable.

Equation (3.10) is rewritten as

$$\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - (\boldsymbol{M}\mathbf{x})_i)^2 \,, \quad \text{or} \quad \min_{\mathbf{x}} ||\boldsymbol{M}\mathbf{x} - \mathbf{y}||_2^2 \,, \tag{3.11}$$

where $(\boldsymbol{M}\mathbf{x})_i$ is the $i-$th component of the vector $\boldsymbol{M}\mathbf{x}$ and $y_i,\ i = 1, \ldots, N$ are the recorded signals at $t_i,\ i = 1, \ldots, N$. If $A(t_i) = S(\mathbf{x}\,, \boldsymbol{\theta}\,, t_i)\,$, vector $\mathbf{x} \in \mathbb{R}^{m_1 n+1}$ and $\boldsymbol{M} \in \mathbb{R}^{N\times(m_1 n+1)}$ is a matrix with $m_1 n + 1$ columns and N rows of the form $S(\boldsymbol{\theta}, t_i)\sin(\omega t_i + \tau)$. For $A(t_i) = P(\mathbf{x}\,, t_i)\,$, vector $\mathbf{x} \in \mathbb{R}^{m_2+1}$ and $\boldsymbol{M} \in \mathbb{R}^{N\times(m_2+1)}$ is a matrix with $m_2 + 1$ columns and N rows of the form $P(t_i)\sin(\omega t_i + \tau)$. The least squares solution can be found by solving the normal equations [9] directly if $\boldsymbol{M}$ is a full-rank matrix. In this case, matrix $\boldsymbol{M}^T\boldsymbol{M}$ is known to be nonsingular and well conditioned [7] therefore the solution of the normal equations is unique. There exist various methods for solving the system of normal equations based on the properties of matrix $\boldsymbol{M}$ for instance, Singular Value Decomposition (SVD) [12] and QR decomposition [89]. The least squares method leads to the normal equations

$$(\boldsymbol{M}^T\boldsymbol{M})\mathbf{x} = \boldsymbol{M}^T\mathbf{y}\,, \tag{3.12}$$

where $\mathbf{y} = (y_1, \ldots, y_N)^T \in \mathbb{R}^N$ is a signal segment recorded at N distinct consecutive time moments and its analytical solution is

$$\mathbf{x} = (\boldsymbol{M}^T\boldsymbol{M})^{-1}\boldsymbol{M}^T\mathbf{y}\,. \tag{3.13}$$

The singularity study of matrix $\boldsymbol{M}$ is demonstrated in Chapter 4 in order to find the best method to solve LLSOM1.

3.5.2 Linear Least Squares Optimization Model 2 (LLSOM2)

In this model, we approach the problem differently. We assume that the wave W_1 defined in (3.7) is shifted vertically by a spline function when $A(t) = S(\mathbf{x}_1\,, \boldsymbol{\theta}\,, t)$ and by a polynomial function when $A(t) = P(\mathbf{x}_1\,, t)$ while in LLSOM1 the wave is oscillating around "zero". Therefore

$$W_2 = \begin{cases} W_1 + S(\mathbf{x}_2\,, \boldsymbol{\theta}\,, t_i) & \text{if} \quad A(t_i) = S(\mathbf{x}_1\,, \boldsymbol{\theta}\,, t_i)\,, \\ W_1 + P(\mathbf{x}_2\,, t_i) & \text{if} \quad A(t_i) = P(\mathbf{x}_1\,, t_i)\,. \end{cases} \tag{3.14}$$

S and P are described in (3.8). Then,

$$\mathbf{x} = \begin{cases} [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^{2m_1n+2}\,, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{m_1n+1}\,, & \text{if} \quad A(t_i) = S(\mathbf{x}_1\,, \boldsymbol{\theta}\,, t_i)\,, \\ [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^{2m_2+2}\,, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{m_2+1}\,, & \text{if} \quad A(t_i) = P(\mathbf{x}_1\,, t_i)\,, \end{cases} \tag{3.15}$$

where m_1 and m_2 are the degrees of S and P respectively. Hence, the corresponding optimization problem is

$$\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - W_2)^2\,, \tag{3.16}$$

where $y_i,\ i = 1, \ldots, N$ are the signal readings at t_i for $i = 1, 2, \ldots, N$ and vector $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2]$ is the spline or the polynomial parameters.

Remark. In our numerical experiments, vertical shift splines $S(\mathbf{x}_2\,, \boldsymbol{\theta}\,, t_i)$ and amplitudes $A(t_i) = S(\mathbf{x}_1\,, \boldsymbol{\theta}\,, t_i)$ have the same degrees and knot locations. In addition, the amplitudes $A(t_i) = P(\mathbf{x}_1\,, t_i)$ and vertical shift polynomials $P(\mathbf{x}_2\,, t_i)$ have the same degrees. Based on the particular application, the amplitude and the vertical shift functions can have different degrees and knot locations. Further, one can apply a spline as an amplitude and a polynomial as a vertical shift and vice versa.

The dimension of this problem depends on the type of approximation chosen for the amplitude. If S is selected as the amplitude then, the dimension is $2m_1n + 2$ (same degree and knots for the vertical shift and the amplitude) whereas if P is selected, the dimension is $2m_2 + 2$ (same degree for the vertical shift and the amplitude). The optimization problem (3.16) is reformulated as

$$\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - (\boldsymbol{B}\mathbf{x})_i)^2 \,, \quad \text{or} \quad \min_{\mathbf{x}} ||\boldsymbol{B}\mathbf{x} - \mathbf{y}||_2^2 \,, \tag{3.17}$$

where $(\boldsymbol{B}\mathbf{x})_i$ is the $i-$th component of the vector $\boldsymbol{B}\mathbf{x}$ and $y_i,\ i = 1, \ldots, N$ are the recorded signals at $t_i,\ i = 1, \ldots, N$. If a spline function is approximated as an amplitude then, vector $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2] \in \mathbb{R}^{2m_1n+2}$ and $\boldsymbol{B} \in \mathbb{R}^{N\times(2m_1n+2)}$ is a matrix with $2m_1n + 2$ columns and N rows of the form $S(\boldsymbol{\theta}, t_i)\sin(\omega t_i + \tau) + S(\boldsymbol{\theta}, t_i)$. If a polynomial function is approximated as an amplitude then, vector $\mathbf{x} \in \mathbb{R}^{2m_2+2}$ and $\boldsymbol{B} \in \mathbb{R}^{N\times(2m_2+2)}$ is a matrix with $2m_2 + 2$ columns and N rows of the form $P(t_i)\sin(\omega t_i + \tau) + P(t_i)$. Optimization problem (3.16) is an LLSP.

The singularity analysis of matrix $\boldsymbol{B}$ from (3.17) is presented in Chapter 4. This analysis leads to find the best solution method for LLSOM2.

3.5.3 Uniform Optimization Model 1 (UOM1)

The wave is approximated similar to W_1 defined in (3.7). In this model, we are minimizing the maximum of the absolute deviation between the original data y_i and the approximated wave. Therefore the corresponding optimization problem is

$$\min_{\mathbf{x}} \quad \max_{i=1,\ldots,N} |y_i - A(t_i)\sin(\omega t_i + \tau)| \,, \tag{3.18}$$

where $y_i,\ i = 1, \ldots, N$ are the signal records at moments t_i for $i = 1, 2, \ldots, N$ and

$$A(t_i) = \begin{cases} S(\mathbf{x}\,, \boldsymbol{\theta}\,, t_i)\,, & \mathbf{x} \in \mathbb{R}^{m_1n+1}\,, \\ P(\mathbf{x}\,, t_i)\,, & \mathbf{x} \in \mathbb{R}^{m_2+1}\,. \end{cases} \tag{3.19}$$

The optimization problem (3.18) has $m_1n + 1$ and $m_2 + 1$ variables with N equality constraints subject to the approximations of $A(t_i)$ defined in (3.19). One can prove the following proposition.

Proposition 3.5.1. *The objective function in UOM1 is a convex function.*

Proof of Proposition 3.5.1. Let $f_i(\mathbf{x}) = y_i - A(t_i)\sin(\omega t_i + \tau)$ for $i = 1, \ldots, N$. Obviously

$$|f_i(\mathbf{x})| = \max\{f_i(\mathbf{x}), -f_i(\mathbf{x})\},$$

and the maximum of linear functions is a convex function. Therefore the objective function in UOM1 is a maximum of a finite number of convex functions and hence, it is convex. □

3.5.4 Uniform Optimization Model 2 (UOM2)

Similar to UOM1, UOM2 is based on uniform approximation. In this model the EEG signal is approximated as W_2 in (3.14) therefore the optimization model is

$$\min_{\mathbf{x}} \quad \max_{i=1,\ldots,N} |y_i - W_2|, \tag{3.20}$$

where $y_i,\ i = 1, \ldots, N$ are the signal records and

$$\mathbf{x} = \begin{cases} [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^{2m_1 n+2}, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{m_1 n+1}, & \text{if} \quad A(t_i) = S(\mathbf{x}_1, \boldsymbol{\theta}, t_i), \\ [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^{2m_2+2}, \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{m_2+1}, & \text{if} \quad A(t_i) = P(\mathbf{x}_1, t_i). \end{cases} \tag{3.21}$$

The dimension of this problem is $2m_1 n + 2$ and $2m_2 + 2$ subject to the selection of S and P respectively as amplitude approximations. The objective functions in (3.20) are convex similar to those in (3.18). The proof of the convexity of

$$|f_i(\mathbf{x})| = |y_i - W_2|,$$

is similar to Proposition 3.5.1.

3.6 Solution Methods

Although LLSOM1 and LLSOM2 are convex quadratic problems, they are not necessarily strictly convex as their Hessian matrices may not be positive definite. There are a variety of methods for solving such problems, but not all methods are equal. Some of them are very efficient, but require strict convexity. Others are more robust, but also less efficient.

Therefore if it is known that the system matrix is non-singular then, the most popular approach for solving the corresponding problem is based on the direct solving of the normal equations defined in (3.12). We refer to the direct method of solving the system of normal equations as the normal equations method. This method is very efficient, fast and accurate when working with non-singular matrices. If the system matrix is singular, the system of normal equations (3.12) has infinitely many solutions and therefore the corresponding optimization problem has more than one solution. There exist other methods for solving the system of normal equations (3.12) such as QR decomposition and SVD.

According to the literature [9, 12, 89], an SVD is more robust and reliable than the normal equations method for rank-deficient or nearly rank-deficient problems. However, this method is substantially more expensive.

An SVD applies orthogonal transformations to reduce the problem to a diagonal system. A square matrix $\boldsymbol{U}$ is orthogonal if $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and an $m \times n$ matrix $\boldsymbol{\Sigma}$ is diagonal if the entries outside the main diagonal are all zero. Singular value decomposition of an $N \times (2m_1n + 2)$ matrix $\boldsymbol{B}$ when S is approximated as the amplitude has form $\boldsymbol{B} = \boldsymbol{U\Sigma V}^T$ where $\boldsymbol{U}$ is an $N \times N$ orthogonal matrix, $\boldsymbol{V}$ is an $(2m_1n+2) \times (2m_1n+2)$ orthogonal matrix and $\boldsymbol{\Sigma}$ is an $N \times (2m_1n+2)$ diagonal matrix with

$$\sigma_{ij} = \begin{cases} 0\,, & \text{for} \quad i \neq j\,, \\ \sigma_i \geq 0\,, & \text{for} \quad i = j\,. \end{cases} \tag{3.22}$$

The diagonal entries σ_i, called singular values of $\boldsymbol{B}$, are usually ordered so that

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{2m_1n+2}. \tag{3.23}$$

Columns $u_i, i = 1, 2, \ldots, N$ of $\boldsymbol{U}$ and $v_i, i = 1, 2, \ldots, 2m_1n + 2$ of $\boldsymbol{V}$ are called left and right singular vectors respectively. The solution corresponds to $\mathbf{x} = \boldsymbol{V\Sigma}^{-1}\boldsymbol{U}^T\mathbf{y}$, where $\mathbf{y} = [y_i]_{i=1}^N$. The preference of the SVD lies in the fact that it always exists and can be computed stably. The computed SVD will be well conditioned because matrices preserve the 2-norm. The flowchart of LLSOM1–2 are presented in Figure 3.2 where LLSOM1 and LLSOM2 are applied as preprocessing approaches to extract the essential features of an EEG signal.

As a consequence, the normal equations method is much faster than an SVD method and requires the strict convexity to be very efficient, but it is not very robust when the system matrix is singular. So, the development of a singularity verification rule is necessary for choosing a better suited approach for solving the system of normal equations. In the next chapter, we demonstrate how this issue can be addressed for LLSOM1–2.

Non-smooth convex problems are a large fraction of convex optimization programs [34]. There exist options to solve them including: transformation to easily solved form, approximation by a smooth function, development of a specific solver and utilization of a subgradient-based method [34]. UOM1 and UOM2 are non-smooth convex problems.

CVX is a MATLAB-based modeling system for solving convex optimization problems. It turns into an optimization modeling language, allowing constraints and objectives to be specified using standard MATLAB expression syntax. The description of the methods can be found on the CVX website [22].

The optimization problems (3.18) and (3.20) can be solved using the CVX software [22, 33]. CVX employs its default and professional solvers called SDPT3 [88, 90] and GUROBI [39] respectively to solve UOM1 and UOM2. The algorithm implemented in SDPT3 is a primal-dual interior point algorithm that uses the infeasible path-following algorithms [90] for solving semidefinite quadratic linear programming problems. For improved efficiency, SDPT3 solves a dual problem.

In addition, UOM1 and UOM2 can be formulated as linear programming problems (LPPs) and solved through LINPROG syntax in MATLAB or other solvers. The solutions of LPPs for UOM1–2 are optimal. The exact formulations is provided in Chapter 4. Their detailed experimental results is described in Chapter 5. UOM1–2 can be solved using Algorithm 3.1.

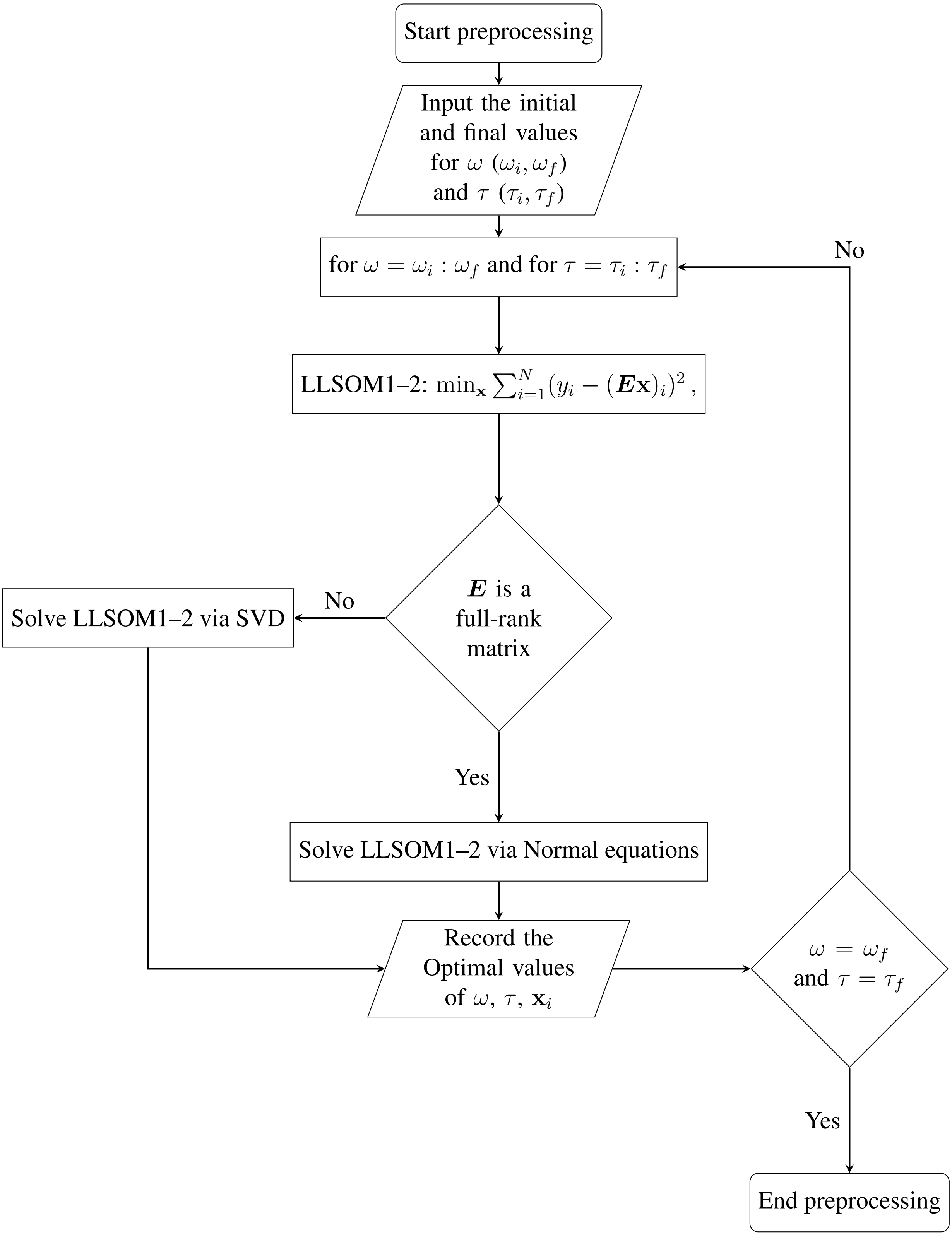
Start preprocessing
Input the initial and final values for ω (ω_i, ω_f) and τ (τ_i, τ_f)
for $\omega = \omega_i : \omega_f$ and for $\tau = \tau_i : \tau_f$
No
LLSOM1–2: $\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - (\boldsymbol{E}\mathbf{x})_i)^2$,
$\boldsymbol{E}$ is a full-rank matrix
No
Solve LLSOM1–2 via SVD
Yes
Solve LLSOM1–2 via Normal equations
Record the Optimal values of ω, τ, $\mathbf{x}_i$
$\omega = \omega_f$ and $\tau = \tau_f$
Yes
End preprocessing

Figure 3.2: LLSOM1–2 Flowchart

Algorithm 3.1 Feature extraction through UOM1–2

```
Specify the initial and final values for the frequency ($\omega_0$ and $\omega_f$) and phase shift ($\tau_0$ and $\tau_f$)
for $\omega = \omega_0 : \omega_f$ and $\tau = \tau_0 : \tau_f$ do
    Set UOM1–2 with fixed $\omega$ and $\tau$.
    Solve the problem through CVX and LINPROG.
    Record the optimal values of the objective function, $\omega$, $\tau$ and $\mathbf{x}$.
end for
```

3.7 Coding Optimization Models

Upon establishing the algorithm to extract the essential features of an EEG signal, the aforementioned optimization models are programmed in MATLAB $R2012b$ and run on a PC with 3.10 GHz CPU and 8 GB of memory. In addition, the CVX professional package containing GUROBI solver is used.

3.8 Classification Algorithms

During the feature extraction stage, the essential features of an EEG signal are extracted through LLSOM1–2 and UOM1–2. In this study, the optimization-based preprocessing approaches (OPAs) are referred to LLSOM1–2 and UOM1–2 that are defined in (3.10), (3.16) and (3.18), (3.20) respectively. Classification algorithms or classifiers identify to which class the extracted features belong. Therefore the efficiency of an OPA to approximate an EEG signal significantly affects the classification accuracy of that signal. If the approximated signal is not accurate enough to describe the original one then, the proposed classifier will have trouble to determine the classes on such extracted features. As a consequence, the development of an OPA in the feature extraction stage is much more important than the development of a classifier in classification problems.

The OPAs reduce the size of classification problem and extract the essential features of an EEG signal. Key features contain the optimal values of objective function, ω, τ and amplitude parameters $\mathbf{x}$ for each segment. To evaluate the performance of an OPA in terms of the classification accuracy of an EEG signal, the classifiers are employed. To obtain the classification accuracy of an EEG signal the classifiers from WEKA [94] are applied on the original dataset and dataset after an OPA. WEKA is an open source data analysis software and its web-site [94] provides all the necessary documentation; therefore a very short description of the classifiers used in this study is provided. The following 12 classifiers are evaluated.

- LibSVM – an integrated software for support vector machines (SVM) classification [94];
- Logistic – a generalized linear model used for binomial regression [94];
- RBF – a classifier that implements a normalized Gaussian radial basis function network, using the K-means clustering algorithm to provide the basis functions [94];
- SMO – a sequential minimal optimization algorithm for training a support vector classifier (a special case of LibSVM) [94];
- Lazy IBK – a K-nearest neighbors classifier (uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance) [94];
- KStar – an instance-based classifier [94];
- LWL – a locally weighted learning classifier that uses an instance based algorithm to assign instance weights [94];
- OneR – a classifier that uses the minimum error attribute for prediction, discretizing numeric attributes [94];
- J48 and J48graft - a classifier based on C4.5 decision tree [94];
- LMT – a logistic model tree based approach, with logistic regression functions at the leaves [94].

All these classifiers are used with their default sets of parameters, except Lazy IBK, which is used with $K = 1$ and 5. Each of the 12 classifiers is trained on the training set and tested on the test set then, the accuracy on the test set is reported. First, the classifiers are used over the original data. The next step is to apply all the above classifiers over the obtained set of features after an OPA.

3.9 Evaluation Criteria

Doctor's decisions rely on diagnosis tests. The characteristics of the diagnosis test can be measured through statistical measurements. The classification accuracy, sensitivity and specificity are widely used metrics to describe a diagnosis test. They are utilized to quantify how reliable a test is. The classification accuracy shows how correct a test detects and excludes a given condition. The sensitivity indicates how good the test is at detection of healthy volunteers while specificity evaluates how likely unhealthy volunteers can be detected correctly. Different statistical measurements help one to interpret the result.

To assess the performance of the OPAs, different statistical measurements such as the classification accuracy (ACC), sensitivity, specificity, false positive rate (FPR) and false negative rate (FNR) are used. They can be derived from a confusion matrix that is detailed below. Therefore the aforementioned metrics are described as follows.

Table 3.1: Description of a confusion matrix.

Actual class	Predicted class	
	non-target	target
non-target	a	b
target	c	d

1. ACC $= (a + d)/(a + b + c + d)$ corresponds to the percentage of correctly classified segments against the total number of tested segments.

2. Sensitivity $= a/(a + b)$ corresponds to the proportion of non-target segments (samples) that have been predicted correctly. This metric is also referred to the Recall value.

3. Precision $= a/(a + c)$ corresponds to the proportion of non-target segments that are truly classified divided by the total number of segments classified as non-target ones.

4. Specificity $= d/(c + d)$ measures the rate of target samples predicted correctly.

5. FPR $= b/(a + b)$ belongs to the rate of non-target samples being categorized as target samples. This measure is also referred to the type-I error.

6. FNR $= c/(c + d)$ belongs to the rate of target samples being categorized as non-target samples. This measure is also referred to the type-II error.

The ACC was used as the criterion to assess the performance of the feature extraction methods in the most of EEG signal analysis applications. According to the numerical experiments, this metric is biased in terms of the unbalanced dataset and the proportions of correctly and incorrectly classified samples. In an unbalanced dataset, a disproportionately large amount of samples belong to a certain class. Therefore the area under a receiver operating characteristic (ROC) curve can be used to assess an OPA where there exists an unbalanced dataset. The ROC curve is produced by plotting the fraction of sensitivity against the fraction of FPR as the threshold for discrimination between two classes is varied [102]. Hence, the area under ROC curve is an important value to evaluate the performance of a classifier regardless of the class distribution. This value lies between 0 and 1. A classifier has an exact discriminative ability if the area under the curve equals 1. In this case, the confusion matrix defined in Table 3.1 has zero values for b and c. It has no discriminating ability when the area is 0.5. A classifier should never have an area less than 0.5 [30]. The area under the ROC curve helps to find the optimal combination of OPAs and classifiers where there exists an unbalanced dataset. In summary, the area under the ROC curve is used to test the quality of a classifier over an unbalanced dataset.

3.10 Evaluating/Testing OPA

In order to evaluate the ability of OPAs to extract the essential features of EEG signals, two types of datasets as numerical experiments will be conducted. The proposed EEG signals were recorded for detection of K-complexes and epileptic seizures. These datasets were suggested by previous researchers in the classification issues for detection of transient events. They will be discussed in the Sections 3.10.1 and 3.10.2.

3.10.1 EEG K-complex Dataset

The K-complex dataset is not publicy available. A qualified doctor should be involed to do data scoring of K-complexes therefore the collection of such datasets is expensive. There are several difficulties in the development of efficient methods for K-complex detection. One of them is data availability since the scoring of K-complexes is not normally kept in PSG analysis records. Another difficulty is that the same segment of data may be scored differently by different scorers [41]. Data scored at Tenon Hospital in Paris are used in this study. On these data, it is more efficient to divide 30 second epochs into 3 equal parts [57] and inspect the presence of K-complexes in each subepoch individually.

Each observation contains a 10 second segment of an EEG recording at a sampling frequency of $100\,\text{Hz}$. Each data segment consists of a sequence of (t_i, y_i), $i = 1, 2, \ldots, 1000$, where y_i is the EEG voltage recorded at time t_i. A dataset with 39 non-K-complexes and 31 K-complexes (70 observations) is used.

As stated in Section 2.5.1 from Chapter 2, K-complexes are specific EEG waveforms [15] and consist of an initial small negative, somewhat sharp wave, followed by a large positive wave. A K-complex in an EEG signal is presented in Figure 3.3 where the amplitude increases sharply and then, returns to the original value. So, one of the main characteristics of K-complexes is a sudden increase in the signal amplitude. They have a wide variety of shapes and are difficult to characterize from other EEG waves due to the fact that EEG signals are non-linear and non-stationary. Therefore the shape of K-complexes is not clear in PSG. The usual method for K-complex detection in order to identify sleep stages is visual (manual) inspection of an EEG signal by a sleep specialist. However, the visual scoring of whole night EEG recordings is a time consuming task [57]. Subsequently, an accurate method for automatic detection of K-complexes is needed.

As a training set a dataset with 30 non-K-complexes and 21 K-complexes (51 observations) is used and a dataset with 9 non-K-complexes and 10 K-complexes (19 observations) is used as a test set. Each of the 12 classifiers defined in Section 3.8 is trained on the training set and tested on the test set then, the accuracy on the test set is reported. It is worthwhile to mention that the EEG K-complex signal contains an unbalanced dataset then, the area under the ROC curve is used to test the quality of classifiers in addition to the ACC. First, the classifiers are used over the original dataset. The next step is to apply all the 12 classifiers over the obtained set of features after an OPA.

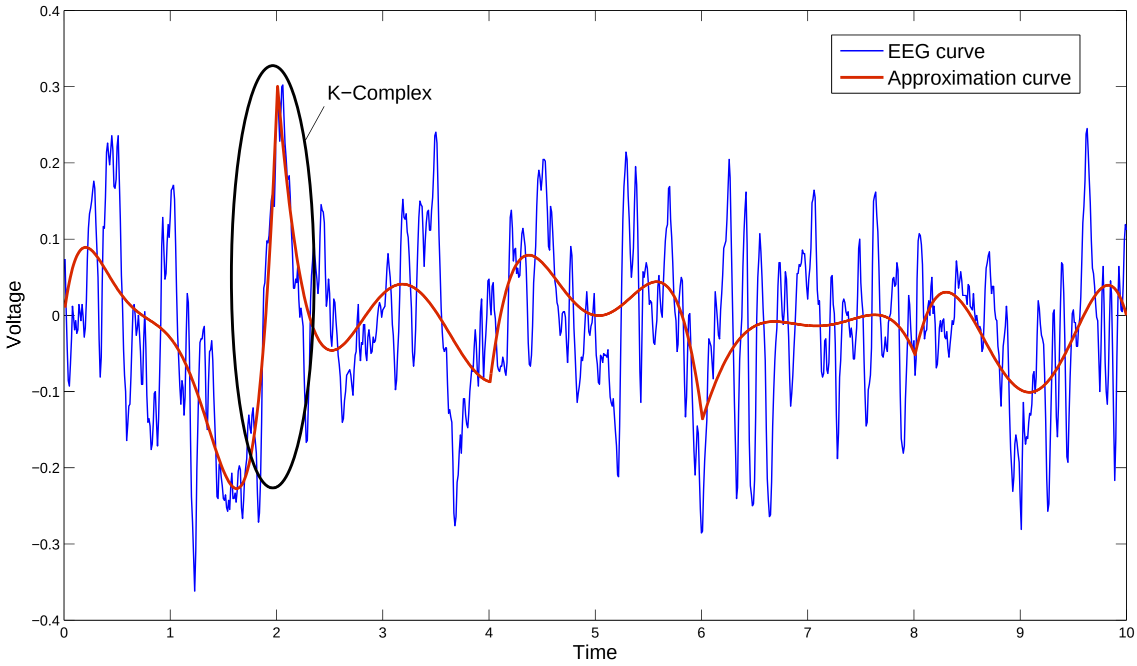

Figure 3.3: The growth of the amplitude in the presence of K-complexes.

3.10.2 EEG Epileptic Dataset

The EEG epileptic dataset used for this study is collected from the epileptic center at the University of Bonn, Germany and studied by Andrzejak *et al.* [3]. This dataset is publicly available and employed to validate the proposed methods. It contains five datasets namely A, B, C, D and E each containing 100 signal channel EEG segments recorded during 23.6 seconds with a sampling frequency of $173.61\,\text{Hz}$ using 12-bit resolution. Band-pass filter settings were $0.53-40\,\text{Hz}$ ($12\,\text{dB/oct}$) therefore each signal has 4097 recordings (a length of 4097 samples).

Segments belong to sets A and B are collected from surface EEG recordings of five healthy volunteers with eyes open and close respectively. Sets C, D and E were obtained from the presurgical diagnosis of five different epileptic patients. EEG recordings of sets C and D were collected during seizure free intervals while set E includes EEG signals during seizure activity. Recordings of A-B,C-D and E datasets were defined as normal, interictal and ictal signals respectively.

Although many preprocessing approaches were tested on sets A and E and achieved high classification accuracy, the effectiveness of different groups of datasets was not investigated thoroughly. It is more desirable to investigate the ability of proposed methods to deal with EEG signals containing different combinations of datasets (A,B,C,D and E). To address this issue, four different binary classification problems are made from aforementioned datasets. All experiments described below are aimed at the detection of epileptic seizure.

- Experiment 1: Classification of sets A, B, C, D (combined) against set E.

  The EEG recordings classified into two different classes. Sets A to D belong to a non-seizure class and set E belongs to a seizure class.

- Experiment 2: Classification of sets A, C, D (combined) against set E.

  Sets A, C and D belong to a non-seizure class and set E belongs to a seizure class. The goal of this experiment is to classify samples of seizure and non-seizure excluding healthy with eyes close.

- Experiment 3: Classification of set B against set E.

  Set B is treated as a non-seizure class while set E as a seizure class.

- Experiment 4: Classification of set A against set E.

  Set A belongs to a non-seizure class and set E belongs to a seizure class.

Proper balancing of datasets where there exists the equal number of segments for each class is necessary to avoid inconsistency in EEG signals and improve the performance of classification algorithms. Therefore the datasets described in [3] are balanced and used to validate the OPAs in the identification of epileptic seizures. Table 3.2 describes the datasets belong to the corresponding experiments concisely.

EEG recordings have five datasets (A, B, C, D and E) each containing 100 segments. They are divided into two classes called non-seizure and seizure for each experiment. In order to balance the datasets, 100 segments (observations) are assigned to each class. These 100 segments are selected as follows.

- Experiment 1: Set E belongs to a seizure class and contains 100 segments therefore the non-seizure class should have 100 segments such that sets A, B,

Table 3.2: Description of the datasets belongs to the experiments.

Experiment #	Classes		Segments		Total Channels
	Non-seizure	Seizure	Non-seizure	Seizure	
1 (ABCD-E)	ABCD	E	100	100	200
2 (ACD-E)	ACD	E	100	100	200
3 (B-E)	B	E	100	100	200
4 (A-E)	A	E	100	100	200

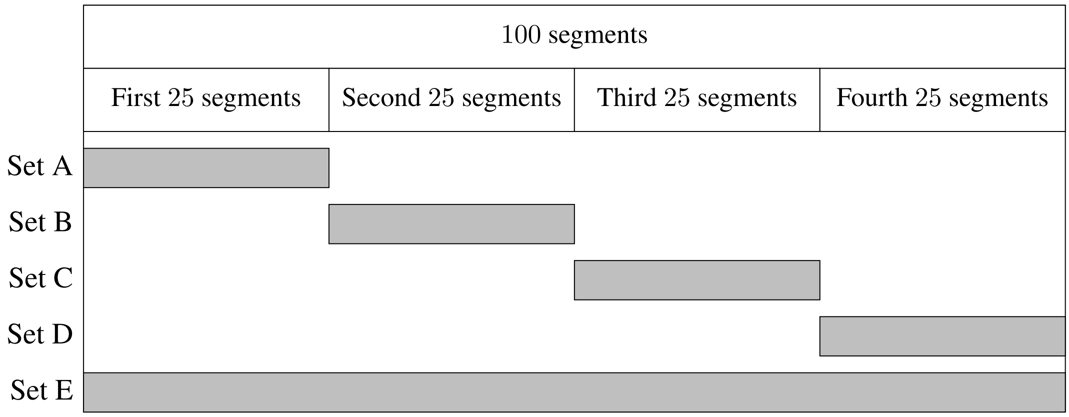

Figure 3.4: Segments selection of Experiment 1 for datasets balancing.

C and D have 25 segments each. Then, the first 25 segments are assigned to set A, the second 25 segments are selected for set B, the third 25 segments are selected for set C and finally the fourth 25 segments are assigned to set D. Figure 3.4 shows how these 25 segments are selected. As it can be seen the last 100 segments are selected for the seizure class containing set E.

- Experiment 2: Similar to Experiment 1, the last 100 segments are selected for the seizure class (set E). Therefore the non-seizure class should have 100 segments such that sets A, C and D have 33, 33 and 34 segments respectively. Then, the first 33 segments are selected for set A, the second 33 segments are assigned to set C and the remaining 34 segments are selected for set D. Figure 3.5 illustrates the way these segments are selected.

- Experiment 3: The second and the last 100 segments are selected for the non-seizure (set B) and the seizure (set E) classes respectively.

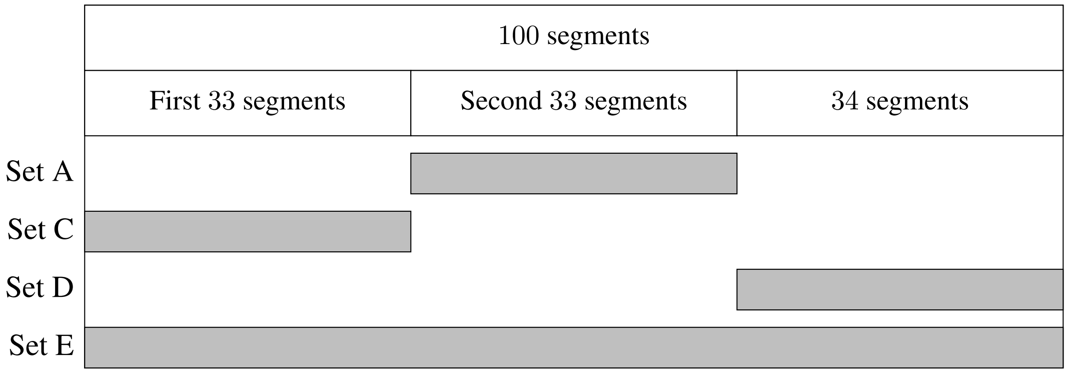

Figure 3.5: Segments selection of Experiment 2 for datasets balancing.

- Experiment 4: The first and the last 100 segments are selected for the non-seizure (set A) and the seizure (set E) classes respectively.

Each dataset is partitioned into training and test sets. Ninety percent of the dataset is selected as a training set and the remaining 10% is for test set. Therefore all experiments, except the first one, that are introduced in Section 3.10.2 have 180 and 20 segments for training and test sets respectively. Further, Experiment 1 has 178 and 22 segments for training and test sets respectively. So, each of 12 classifiers from WEKA is trained on the training set and tested on the test set and the classification accuracy on the test set is reported. The classifiers are employed over the original dataset and the obtained set of features after OPAs.

3.11 Summary

In this chapter, the development of novel feature extraction methods for classification of two categories of EEG signals in presence of K-complexes and seizures are presented. Moreover, the steps in conducting this study have been described. Generally, the methodology commences with an EEG signal modeling followed by

- the signal amplitude approximation,
- convex optimization models as feature extraction methods,
- solution methods,
- coding optimization models,

- classification algorithms,
- evaluation criteria and
- evaluating/testing OPA.

The developed approach introduces LLSOM1–2 and UOM1–2 for feature extraction and 12 classifiers described in Section 3.8 for binary classification of two types of EEG signals. To validate the developed approach two datasets namely K-complex and epileptic seizure datasets from different types of EEG signals are used. Further analysis of LLSOM1–2 and UOM1–2 will be presented in Chapter 4.

Chapter 4

Mathematical Study of Feature Extraction Methods and Numerical Implementations

4.1 Introduction

In this chapter, an extensive mathematical analysis of linear least squares problems (LLSPs) is provided. This analysis leads to the development of singularity verification rules. These rules are used to choose a better suited method for solving LLSPs. Four optimization problems frequently appearing in approximation and signal processing are considered. These problems, namely Model 1, Model 2, Model 3 and Model 4, are used as feature extraction methods to select the key features of EEG signals. Models 1 and 2 are formulated as LLSPs while Models 3 and 4 are formulated as uniform approximation problems that are non-smooth convex. In addition, linear programming reformulations of uniform approximation problems are provided to solve such non-smooth convex optimization problems and verify the optimal solutions.

The signal approximations are constructed as products of two functions: polynomial spline (piecewise polynomial) and a prototype (also called basis) function defined by the application. Common examples of prototype functions are sine and cosine functions. In Models 1 and 3, the wave oscillates around “zero level” while Models 2 and 4 admit a vertical shift (signal biasing). This shift is modeled as another polynomial spline.

The choice of polynomial splines is due to the fact that these functions combine the simplicity of polynomials and the flexibility, which is achieved by switching from one polynomial to another. Therefore on the one hand, the corresponding optimization problems can be solved efficiently and, on the other hand, the approximation inaccuracy is low. The sum of the corresponding deviation squares is referred to the approximation inaccuracy.

Model 1 and Model 2 are LLSPs. There are a variety of methods to solve such problems [10, 32, 53]. If it is known that the system matrix is non-singular, then, the most popular approach for solving the corresponding LLSP is based on the direct solving of the normal equations (unique solution for non-singular systems). This method is very efficient (fast and accurate) when working with non-singular matrices. If the matrix is singular, one needs to apply more robust methods, for example, QR decomposition or Singular Value Decomposition (SVD). These methods are much more computationally expensive.

Necessary and sufficient conditions for non-singularity of Model 1, and sufficient conditions for non-singularity and singularity of Model 2 are developed. In most applications, the number of columns of the corresponding system matrix is lower than the number of rows. Therefore for singularity verification one needs to verify that the number of linearly independent rows is the same as the number of columns.

Truncated power basis functions are used to define polynomial splines. Another way to construct basis functions is through B-splines. B-splines have several computational advantages when running numerical experiments (have smallest possible support) [63]. However, truncated power functions are occasionally more convenient when theoretical properties of the models are the objectives [63].

4.2 Polynomial Splines

A polynomial spline is a piecewise polynomial of a specified degree. The points where the polynomial pieces join each other are called spline knots. Polynomial splines combine the simplicity of polynomials and an additional flexibility property that enables them to change abruptly at the points of switching from one polynomial to another (spline knots). These attributes are essential therefore on one hand the corresponding optimization problems are relatively inexpensive to solve and, on the other hand, the obtained approximations are accurate enough for reflecting the essential characteristics of the original signal (raw data). Therefore polynomial splines are commonly used in approximation problems [56, 63, 76]. Similar to Equation (3.4) described in Chapter 3, polynomial splines are constructed through a truncated power function [63]:

$$S(\mathbf{x}\,,\boldsymbol{\theta}\,,t) = x_0 + \sum_{j=1}^{m} x_{1j}t^j + \sum_{l=2}^{n}\sum_{j=1}^{m} x_{lj}((t-\theta_{l-1})_+)^j\,, \tag{4.1}$$

where m is the spline degree, n is the number of subintervals, N is the number of recordings in a time segment $[t_1, t_N]$ when the original signal is recorded, $\mathbf{x} = (x_0, x_{11}, \dots, x_{mn}) \in \mathbb{R}^{mn+1}$ are the spline parameters and $\boldsymbol{\theta} = (\theta_1, \dots, \theta_{n-1})$ are the spline knots such that

$$t_1 \le \theta_1 \le \theta_2 \le \dots \le t_N\,.$$

In some cases, the border points $\theta_0 = t_1$ and $\theta_n = t_N$ are also considered as knots. Furthermore, for simplicity, we refer to segments $[\theta_0, \theta_n]$ as intervals and to subsegments $[\theta_{i-1}, \theta_i], i = 1, 2, \dots, n$ as subintervals. The presentation shown in (4.1) implies that the polynomial splines are continuous. This is achieved by omitting the constant terms in the polynomial presentations for all the subintervals starting from the second one. In general, polynomial splines can be discontinuous at their knots [63]. However, we only consider continuous polynomial splines.

The spline knots can be free or fixed. If the knots are free, then, they are considered as additional variables in the set of decision variables and thus, $\mathbf{x} = (x_0, x_{11}, \dots, x_{mn}, \boldsymbol{\theta})$. In this case, the corresponding optimization problem becomes more complex [11, 63, 64, 83, 84]. In particular, it becomes non-convex. Generally, it is much easier to solve a higher dimension fixed knots problem (with a considerably larger number of subintervals) than a free knots one.

In most applications, we have been working with the signal that is recorded at the same sampling frequency (for example, $100\,\mathrm{Hz}$ means 100 recordings per second).

The knots split the original interval into subintervals and a simple equidistant knots distribution is used. In this case,

$$\theta_{l-1} = t_0 + (t_N/n)(l-1),\ l = 2, \ldots, n\,. \tag{4.2}$$

Therefore each subinterval contains the same number of signal recordings. However, in some applications different number of recordings per subinterval can be used. This flexibility is incorporated in the singularity conditions developed further in this chapter.

4.3 Linear Least Squares-based Models

Two types of models are considered such that first model corresponds to the case when the wave is oscillating around "zero level" while the second one enables a vertical shift (signal biasing) in a form of a polynomial spline. Consider a signal segment $\mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$ where $y_i,\ i = 1, \ldots, N$ are evaluated at time $t_i,\ i = 1, \ldots, N$. This signal is approximated by a function $f(t)$ from one of the following two models:

$$\text{Modeling function 1}: \ f(t) = S(\mathbf{x}, t)g(t)\,, \tag{4.3}$$

and

$$\text{Modeling function 2}: \ f(t) = S_1(\mathbf{x}_1, t)g(t) + S_2(\mathbf{x}_2, t)\,, \tag{4.4}$$

where $S(\mathbf{x}, t),\ S_1(\mathbf{x}_1, t)$ and $S_2(\mathbf{x}_2, t)$ are polynomial splines with fixed knots. $\mathbf{x} = (x_0, x_{11}, \ldots, x_{mn}) \in \mathbb{R}^{mn+1}$, $\mathbf{x}_1 \in \mathbb{R}^{mn+1}$ and $\mathbf{x}_2 \in \mathbb{R}^{mn+1}$ where m is the spline degree and n is the number of subintervals. Vectors of $\mathbf{x}$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the corresponding spline parameters and $g(t)$ is a prototype function (for example, sine, cosine). Both the spline and prototype functions can be defined in the whole segment of $[t_1, t_N]$. However, we are only considering discrete time moments $t_i, i = 1, 2, \ldots, N$ when the signal was recorded. Therefore a prototype function defined only in the time moments t_i can be used. To summarize, the function product $S(\mathbf{x}, t)g(t)$ has to be defined only at the discrete time moments t_i.

In most cases, the choice of the prototype functions is application driven (specified by application experts, for example, engineers). The properties of S in (4.3) are described in (4.1). The structures of S_1 and S_2 in (4.4) are similar to S. In addition, we refer to $S(\mathbf{x},t)$ and $S_1(\mathbf{x}_1,t)$ as “main splines” (since they form the shape of the corresponding approximations); and to $S_2(\mathbf{x}_2,t)$ as “shift spline” (signal biasing). Note that in Model 1, the corresponding shift spline is zero.

These two approximation models can be formulated as linear least squares problems:

$$\text{Model 1}: \min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - S(\mathbf{x},t_i)g(t_i))^2 \,, \tag{4.5}$$

and

$$\text{Model 2}: \min_{\mathbf{x_1},\mathbf{x_2}} \sum_{i=1}^{N} (y_i - S_1(\mathbf{x}_1,t_i)g(t_i) - S_2(\mathbf{x}_2,t_i))^2 \,. \tag{4.6}$$

In the following two sections these models are studied in depth.

4.3.1 Model 1

Assume that the spline degree is m, the number of subintervals is n and the corresponding knots are

$$t_0 = \theta_0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_{n-1} \le \theta_n = t_N. \tag{4.7}$$

Model 1 is an LLSP since it can be rewritten as follows:

$$\min_{\mathbf{x}} \|\boldsymbol{M}\mathbf{x} - \mathbf{y}\|_2^2 \,, \tag{4.8}$$

where $\mathbf{y} = (y_1,\dots,y_N)^T \in \mathbb{R}^N$, $y_i,\ i = 1,\dots,N$ are the recorded signal values at $t_i,\ i = 1,\dots,N$ and $\mathbf{x} \in \mathbb{R}^{mn+1}$. Matrix $\boldsymbol{M} \in \mathbb{R}^{N\times(mn+1)}$ has N rows of the form $S(\boldsymbol{\tau},t_i)g(t_i)$ and $mn+1$ columns where m is the degree of spline and n is the number of subintervals. It can be expressed as $\boldsymbol{M} = \boldsymbol{D}\boldsymbol{V}$ where $\boldsymbol{V} = [\boldsymbol{V_1} \quad \boldsymbol{V_2} \cdots \boldsymbol{V_n}]$,

$$\boldsymbol{V_1} = \begin{bmatrix} 1 & t_1 & \dots & t_1^m \\ 1 & t_2 & \dots & t_2^m \\ \vdots & \vdots & \ddots & \vdots \\ 1 & t_N & \dots & t_N^m \end{bmatrix} \in \mathbb{R}^{N\times(m+1)} \,, \tag{4.9}$$

$$\boldsymbol{V_j} = \begin{bmatrix} \beta_{1,j-1} & \dots & \beta_{1,j-1}^m \\ \beta_{2,j-1} & \dots & \beta_{2,j-1}^m \\ \vdots & \ddots & \vdots \\ \beta_{N,j-1} & \dots & \beta_{N,j-1}^m \end{bmatrix} \in \mathbb{R}^{N\times m}\,,\ \text{for}\ j = 2,\dots,n\,, \tag{4.10}$$

and

$$\boldsymbol{D} = \begin{bmatrix} g(t_1) & 0 & \dots & 0 \\ 0 & g(t_2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & g(t_N) \end{bmatrix} \in \mathbb{R}^{N\times N}\,. \tag{4.11}$$

In the aforementioned matrices $\beta_{i,j-1} = \max\{0, t_i - \theta_{j-1}\}$ are the truncated power functions, $j = 2,\dots,n,\ i = 1,\dots,N$ and $g(t_i)$ for $i = 1,\dots,N$ are the prototype functions. Note that matrix $\boldsymbol{M}$ is a lower block triangular matrix since $\beta_{i,j-1} = 0$ when $t_i \leq \theta_{j-1}$.

An LLSP's solution can be found by solving the normal equations directly if $\boldsymbol{M} \in \mathbb{R}^{N\times(mn+1)}$ is a full-rank matrix. In this case, the least squares solution is unique. In the rest of this chapter, the direct method for solving the system of normal equations is referred to the normal equations method. There are other methods for solving systems of normal equations such as QR decomposition and SVD [9, 12, 89].

The least squares method leads to the system of normal equations:

$$(\boldsymbol{M}^T\boldsymbol{M})\mathbf{x} = \boldsymbol{M}^T\mathbf{y}\,. \tag{4.12}$$

The normal equations method is much faster than QR decomposition or SVD. However, it is not very robust when matrix $\boldsymbol{M}$ is singular. Therefore the development of a singularity verification rule is essential for choosing a better suited approach for solving the system of normal equations.

4.3.2 Model 2

In this model, we assume that the wave (signal) described in Equation (4.3) is shifted vertically (signal biasing) by a spline function. Similar to Model 1, the spline degree is m, the number of subintervals is n and the corresponding knots are

$$t_0 = \theta_0 \le \theta_1 \le \theta_2 \le \cdots \le \theta_{n-1} \le \theta_n = t_N \,.$$

Then, the corresponding optimization problem is an LLSP and can be formulated as follows:

$$\min_{\mathbf{x}} \|\boldsymbol{B}\mathbf{x} - \mathbf{y}\|_2^2 \,, \tag{4.13}$$

where $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}] \in \mathbb{R}^{2mn+2}$, $\mathbf{x}_1$ and $\mathbf{x}_2$ are described in Equation (4.4) and $\mathbf{y} \in \mathbb{R}^N$ is the original signal (see (4.6) for details). Matrix $\boldsymbol{B} \in \mathbb{R}^{N \times (2mn+2)}$ has N rows of the form $S_1(\boldsymbol{\tau}, t_i)g(t_i) + S_2(\boldsymbol{\tau}, t_i)$ and $2mn + 2$ columns. So, the system matrix of the optimization model (4.6) is

$$\boldsymbol{B} = [\boldsymbol{DV} \quad \boldsymbol{V}] = [\boldsymbol{M} \quad \boldsymbol{V}], \tag{4.14}$$

where $\boldsymbol{V}$ and $\boldsymbol{D}$ are described in (4.9), (4.10) and (4.11) respectively.

One can rearrange the columns of matrix $\boldsymbol{B}$ in such a way that the updated matrix contains zero blocks in the top-right corner. This can be achieved by grouping the columns that correspond to the same subinterval together. Note that the first sub-block of the updated matrix $\widetilde{\boldsymbol{B}}$ contains $2m + 2$ columns and all the remaining sub-blocks contain $2m$ columns. Then, the updated matrix $\widetilde{\boldsymbol{B}}$ can be expressed as

$$\widetilde{\boldsymbol{B}} = [\widetilde{\boldsymbol{B}}_{\mathbf{11}}\widetilde{\boldsymbol{B}}_{\mathbf{21}} \quad \widetilde{\boldsymbol{B}}_{\mathbf{12}}\widetilde{\boldsymbol{B}}_{\mathbf{22}} \quad \dots \quad \widetilde{\boldsymbol{B}}_{\mathbf{1}\boldsymbol{n}}\widetilde{\boldsymbol{B}}_{\mathbf{2}\boldsymbol{n}}] \,, \tag{4.15}$$

where the first index of $\widetilde{\boldsymbol{B}}_{\boldsymbol{ij}}$ for $i = 1, 2$ indicates the main ($i = 1$) and shift ($i = 2$) splines respectively while the second index ($j = 1, \dots, n$) determines the $j-$th subinterval. The first sub-block ($\widetilde{\boldsymbol{B}}_{\mathbf{11}}\widetilde{\boldsymbol{B}}_{\mathbf{21}}$) contains $2m + 2$ columns and the following ones ($\widetilde{\boldsymbol{B}}_{\mathbf{1}\boldsymbol{j}}\widetilde{\boldsymbol{B}}_{\mathbf{2}\boldsymbol{j}}$) for $j = 2, \dots, n$ contain $2m$ columns. In the rest of this chapter, we simplify our notation and substitute “matrix $\widetilde{\boldsymbol{B}}$” by just “matrix $\boldsymbol{B}$”.

According to the numerical experiments, matrix $\boldsymbol{B}$ is a rank-deficient matrix. $\boldsymbol{B}^T\boldsymbol{B}$ is singular. As a consequence, the system of normal equations $(\boldsymbol{B}^T\boldsymbol{B})\mathbf{x} = \boldsymbol{B}^T\mathbf{y}$ has infinitely many solutions. Therefore more robust methods such as QR decomposition or SVD are required to solve the corresponding optimization problem (4.6) [9, 12, 89].

Note that QR decomposition and SVD are substantially more expensive than solving normal equations. Therefore the importance of developing a singularity verification rule to select a proper approach for solving the corresponding optimization problems is clear here. In Section 4.4, we demonstrate how this issue can be addressed.

4.4 Singularity Study of Linear Least Squares-based Models

It has been pointed out that matrices of $\boldsymbol{M}$ and $\boldsymbol{B}$ can be expressed as a block lower triangular matrix $\boldsymbol{A}$ such that

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_{11} & 0 & \cdots & 0 & 0 \\ \boldsymbol{A}_{21} & \boldsymbol{A}_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \boldsymbol{A}_{33} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{n1} & \boldsymbol{A}_{n2} & \boldsymbol{A}_{n3} & \cdots & \boldsymbol{A}_{nn} \end{bmatrix}, \tag{4.16}$$

where $\boldsymbol{A}_{j1},\ j = 1,\dots,n$ has N_j rows (N_j is the total number of time moments assigned to the j-th subinterval), $m+1$ and $2m+2$ columns for matrices of $\boldsymbol{M}$ and $\boldsymbol{B}$ respectively, $\boldsymbol{A}_{jk},\ j,k = 2,\dots,n,\ j \geq k$ has N_j rows, m and $2m$ columns for $\boldsymbol{M}$ and $\boldsymbol{B}$ respectively, and the top-right corner ($\boldsymbol{A}_{jk}$, for $k > j,\ j,k = 1,\dots,n$) contains zeros since

$$\max\{0, t_i - \theta_{l-1}\} = 0,\ \text{for all}\ t_i \leq \theta_{l-1},\ l = 2,\dots,n.$$

In some cases, one or more of the time moments can coincide with the corresponding spline knots. To avoid possible ambiguity we assign the time moments into the subintervals according to the following subdivision rule:

$$[\theta_0, \theta_1],\ (\theta_{l-1}, \theta_l],\ l = 2,\dots n. \tag{4.17}$$

Therefore the first subinterval includes both borders while the remaining subintervals only include the right border.

Note that all the diagonal blocks $\boldsymbol{A}_{ii},\ i = 1, \dots, n$ in (4.16) are rectangular matrices. The number of columns in $\boldsymbol{A}_{11}$ is $m + 1$ and $2m + 2$ for $\boldsymbol{M}$ and $\boldsymbol{B}$ respectively while the number of columns in $\boldsymbol{A}_{ii},\ i = 2, \dots, n$ is m and $2m$ for $\boldsymbol{M}$ and $\boldsymbol{B}$ respectively. The number of rows in $\boldsymbol{A}_{ii},\ i = 1, \dots, n$ coincided with the number of time moments assigned to the $i-$th subinterval (subdivision rule (4.17)).

4.4.1 Model 1

In this section, a necessary and sufficient non-singularity condition is developed for Model 1 (oscillation around "zero"). If this condition is satisfied, we can guarantee that the corresponding matrices are non-singular and therefore one can apply the normal equations method that is well-known to be fast and efficient for solving such problems. So, the following theorem holds (a necessary and sufficient non-singularity condition, Model 1).

Theorem 4.4.1. *Suppose that the spline degree is m, the number of subintervals is n and the corresponding spline knots are*

$$\theta_0 = t_1 \le \theta_1 \le \dots \le \theta_{n-1} \le \theta_n = t_N\,.$$

Matrix $\boldsymbol{A}$ is non-singular if and only if the following inequalities satisfy

$$N_1 - K_1 \ge m + 1\,, \text{ and } N_l - K_l \ge m,\ l = 2, \dots, n, \tag{4.18}$$

where N_l and K_l, $l = 1, \dots, n$ are the total number of recordings in $l-$th subinterval described in (4.17) and the number of time moments t_i in the $l-$th subinterval when $g(t_i) = 0\,, i = 1, \dots, N$ respectively.

Proof of Theorem 4.4.1. The proof is based on two important facts.

1. For a rectangular matrix changing the order of the rows or multiplying a row by non-zero constants (elementary row operations) do not change the rank of the matrix.

2. The determinant of a square Vandermonde matrix

$$\boldsymbol{V} = \begin{pmatrix} 1 & X_1 & X_1^2 & \ldots X_1^{m-1} \\ 1 & X_2 & X_2^2 & \ldots X_2^{m-1} \\ 1 & X_3 & X_3^2 & \ldots X_3^{m-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & X_m & X_m^2 & \ldots X_m^{m-1} \end{pmatrix}, \tag{4.19}$$

can be expressed as

$$\det(\boldsymbol{V}) = \prod_{1\le i<j\le m} (X_j - X_i)\,, \tag{4.20}$$

and therefore it can not be zero if all $X_i,\ i = 1,\ldots,m$ are distinct.

Note that the rows of the diagonal block $\boldsymbol{A}_{11}$ of (4.16) are Vandermonde matrix rows that are multiplied by a constant $g(t_j),\ j = 1,\ldots,N_1$, where N_1 is the number of time moments assigned to the first subinterval. Therefore it is possible to extract $m+1$ linearly independent rows from the first N_1 rows of $\boldsymbol{A}$ if $N_1 - K_1 \ge m+1$. For the second subinterval the situation is similar but each row is multiplied by

$$g(t_j) \times (t_j - \theta_1),\ j = N_1+1,\ldots,N_1+N_2\,. \tag{4.21}$$

Since for the second subinterval none of the time moments can coincide with θ_1 one can conclude that

$$(t_j - \theta_1) > 0,\ j = N_1+1,\ldots,N_1+N_2\,. \tag{4.22}$$

Therefore it is possible to extract m linearly independent rows from the block of rows $N_1+1,\ldots,N_1+N_2$ of $\boldsymbol{A}$ if $N_2 - K_2 \ge m$. These rows will be also linearly independent with the $m+1$ rows extracted from the first N_1 rows of $\boldsymbol{A}$. By continuing the process, we finally have $mn+1$ linearly independent rows and therefore matrix $\boldsymbol{A}$ is a full-rank matrix. Suppose now that the matrix $\boldsymbol{A}$ is full-rank. This means that there are $mn+1$ linearly independent rows. We have to prove that the first diagonal block of matrix $\boldsymbol{A}$ contains at least $m+1$ linearly independent rows and all the other diagonal blocks have at least m linearly independent rows. If one of the diagonal blocks contains less rows than it is required, then, there is at least one block that contains more rows. In this case, these extra rows will be linearly dependent with the rest of rows from the same block. This contradiction proves the theorem. □

Generally, it is not always easy to estimate $K_l,\ l = 1, \dots, n$. In Section 4.5, we give an example where $g(t)$ is a periodical (sine) function and therefore the corresponding $K_l,\ l = 1, \dots, n$ can be estimated through the corresponding frequencies.

4.4.2 Model 2

In this model, each diagonal block $\boldsymbol{A}_{ii},\ i = 1, \dots, n$ in (4.16) is a double block. The rows of the first sub-block (main spline) can be obtained by multiplying the rows of the second sub-block (shift spline) by the corresponding constants $g(t_j),\ j = 1, \dots, N$.

Consider $\boldsymbol{A}_{11}$ and divide it into two parts: any $m + 1$ rows form the bottom part and the remaining rows form the top part (row permutations do not affect the rank of $\boldsymbol{A}_{11}$). Therefore $\boldsymbol{A}_{11}$ contains four sub-blocks: $\boldsymbol{\Lambda}_{11} \in \mathbb{R}^{(N_1-m-1)\times m+1}$, $\boldsymbol{\Lambda}_{12} \in \mathbb{R}^{(N_1-m-1)\times(m+1)}$, $\boldsymbol{\Lambda}_{21} \in \mathbb{R}^{(m+1)\times(m+1)}$ and $\boldsymbol{\Lambda}_{22} \in \mathbb{R}^{(m+1)\times(m+1)}$. Hence,

$$\boldsymbol{A}_{11} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix}. \tag{4.23}$$

$\boldsymbol{\Lambda}_{22}$ is a full-rank sub-block in $\boldsymbol{A}_{11}$ since it is a Vandermonde-type matrix. By applying equivalent row operations, one can obtain zeros everywhere in sub-block $\boldsymbol{\Lambda}_{12}$ using the last $m + 1$ rows of $\boldsymbol{A}_{11}$. Therefore there exists a unique set of $\gamma^1_{ij},\ j = 1, \dots, N_1 - m - 1,\ i = 1, \dots, m + 1$, such that

$$\boldsymbol{\Lambda}^j_{12} = \sum_{i=1}^{m+1} \gamma^1_{ij} \boldsymbol{\Lambda}^i_{22}\,, \tag{4.24}$$

where $\boldsymbol{\Lambda}^j_{12}$ is the $j-$th row of $\boldsymbol{\Lambda}_{12}$ and $\boldsymbol{\Lambda}^i_{22}$ is the $i-$th row of $\boldsymbol{\Lambda}_{22}$.

The rows of $\boldsymbol{\Lambda}_{11}$ and $\boldsymbol{\Lambda}_{21}$ are denoted by $\boldsymbol{\Lambda}^j_{11}$ and $\boldsymbol{\Lambda}^i_{21}$ respectively. In this case, $j = 1, \dots, N_1 - m - 1$ and $i = 1, \dots, m + 1$. Then, the rows $\tilde{\boldsymbol{\Lambda}}^j_{11}$ for

$j = 1, \ldots, N_1 - m - 1$ of the updated sub-block $\tilde{\mathbf{\Lambda}}_{11}$ (obtained from $\mathbf{\Lambda}_{11}$) are as follows:

$$\tilde{\mathbf{\Lambda}}_{11}^{j} = \mathbf{\Lambda}_{11}^{j} - \sum_{i=1}^{m+1} \gamma_{ij}^{1} \mathbf{\Lambda}_{21}^{i}. \tag{4.25}$$

Note that $\boldsymbol{A}_{11}$ is full-rank if and only if

$$\tilde{\mathbf{\Lambda}}_{11} = \begin{bmatrix} \tilde{\mathbf{\Lambda}}_{11}^{1} \\ \tilde{\mathbf{\Lambda}}_{11}^{2} \\ \vdots \\ \tilde{\mathbf{\Lambda}}_{11}^{N_1-m-1} \end{bmatrix}, \tag{4.26}$$

is full-rank. Similar reasoning can be applied to remaining subintervals, taking into account that

- the dimension of each block $\boldsymbol{A}_{ii},\ i = 2, \ldots, n$ is $N_i \times 2m$, where N_i is the total number of time moments in the $i-$th subinterval and
- the left border point is not included in any of each subinterval.

Hence, the following theorem holds (a sufficient condition for non-singularity, Model 2).

Theorem 4.4.2. *Suppose that the spline degree is m, the number of subintervals is n and the corresponding spline knots are*

$$t_1 = \theta_0 \leq \theta_1 \leq \cdots \leq \theta_{n-1} \leq \theta_n = t_N \, .$$

If it is possible to construct the bottom sub-blocks in each matrix $\boldsymbol{A}_{ii}$ in such a way that the corresponding matrices in (4.26) are full-rank, then, matrix $\boldsymbol{B}$ is also full-rank.

Note that if $g(t)$ is a constant, then, the condition of Theorem 4.4.2 can not be satisfied by choosing appropriate bottom sub-blocks. Moreover, it indicates that matrix $\boldsymbol{B}$ in Model 2 is rank-deficient since there exists a column that is obtained by multiplication of another column by the corresponding $g(t_i),\ i = 1, \ldots, N$.

The main advantage of Theorem 4.4.2 is that one needs to evaluate the ranks of several lower dimensional matrices. There are also two disadvantages.

1. This condition is a sufficient condition but not necessary.
2. There are many ways to form the “bottom sub-blocks” especially when the size of the matrices is large.

Therefore we need to develop optimality conditions that are more practical.

Consider $l-$th subinterval, then, the corresponding time moments are

$$t_i,\ i = N_1 + N_2 + \dots N_{l-1} + 1, \dots, N_1 + N_2 \dots + N_l\,, \tag{4.27}$$

where N_1, N_2 and N_l are the number of signal recordings assigned to the first, second and $l-$th subintervals. Some of the functions values $g(t_i)$ within the same subinterval may coincide. Let $K_l,\ l = 1, \dots, n$ be the multiplicity coefficients for each subinterval that are the highest number of time moments with the same function values. For example, if a subinterval contains five recordings (time moments) and the corresponding function values are the same at three distinct time moments (function values at the remaining time moments may or may not coincide) then, the corresponding multiplicity coefficient is three. Clearly, for the $l-$th subinterval the multiplicity coefficient K_l is a natural number and $1 \leq K_l \leq N_l$.

Theorem 4.4.3. *If*

$$N_1 - m - 1 \geq K_1 \geq m + 1\,, \tag{4.28}$$

and

$$N_l - m \geq K_l \geq m,\ l = 2, \dots n\,, \tag{4.29}$$

where N_l and K_l , $l = 1, \dots n$ are the total number of recordings in l-th subinterval and the multiplicity coefficients for l-th subinterval respectively. Then, matrix $\boldsymbol{B}$ is full-rank.

Proof of Theorem 4.4.3. Consider each diagonal block of matrix $\boldsymbol{A}$ separately. For simplicity assume that for a given subinterval the multiplicity coefficient is $\widetilde{K}$, the number of recordings in each subinterval is $\widetilde{N}$ and the corresponding time moments are $t_1, \ldots, t_{\widetilde{N}}$. Hence, the structure of each diagonal block of matrix $\boldsymbol{A}$ is

$$\boldsymbol{A}_d = \begin{pmatrix} g(t_1)b_1 & b_1 \\ g(t_2)b_2 & b_2 \\ \vdots & \vdots \\ g(t_{\widetilde{N}})b_{\widetilde{N}} & b_{\widetilde{N}} \end{pmatrix}, \tag{4.30}$$

where $b_i,\ i = 1, \ldots, \widetilde{N}$ are the rows of the diagonal submatrices of $\boldsymbol{B}$ that correspond to the time moments $t_i,\ i = 1, \ldots, \widetilde{N}$. It is worth to note that changing the order of columns does not affect the singularity of matrix $\boldsymbol{A}_d$. Then, one can rearrange the columns of matrix $\boldsymbol{A}_d$ such that

$$\boldsymbol{A}_d = \begin{pmatrix} b_1 & g(t_1)b_1 \\ b_2 & g(t_2)b_2 \\ \vdots & \vdots \\ b_{\widetilde{N}} & g(t_{\widetilde{N}})b_{\widetilde{N}} \end{pmatrix}. \tag{4.31}$$

Now we rearrange the time moments in such a way that the first K rows correspond to the time moments with the same function value $g(t_1)$ (function value with the largest number of distinct time moments). Then,

$$\widetilde{\boldsymbol{A}}_d = \begin{pmatrix} b_1 & g(t_1)b_1 \\ b_2 & g(t_1)b_2 \\ \vdots & \vdots \\ b_K & g(t_1)b_K \\ b_{K+1} & g(t_{k+1})b_{K+1} \\ \vdots & \vdots \\ b_{\widetilde{N}} & g(t_{\widetilde{N}})b_{\widetilde{N}} \end{pmatrix} = [\boldsymbol{A}_1\boldsymbol{A}_2], \tag{4.32}$$

where $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are same size matrices. $\widetilde{\boldsymbol{A}}_d$ has the same rank as $\boldsymbol{A}_d$. The following matrix is obtained by multiplying all the columns of $\boldsymbol{A}_1$ by $g(t_1)$ and subtract them from the corresponding columns of $\boldsymbol{A}_2$.

$$\tilde{\tilde{\boldsymbol{A}}}_d = \begin{pmatrix} b_1 & \mathbf{0} \\ b_2 & \mathbf{0} \\ \vdots & \vdots \\ b_K & \mathbf{0} \\ b_{K+1} & (g(t_{K+1}) - g(t_1))b_{K+1} \\ \vdots & \vdots \\ b_N & (g(t_N) - g(t_1))b_N \end{pmatrix}. \tag{4.33}$$

Therefore each diagonal block is a block triangular (lower triangular) matrix with two diagonal sub-blocks. Each sub-block contains Vandermonde-type rows (or Vandermonde-type rows times nonzero constants). If (4.28)-(4.29) are satisfied, then, the number of rows in each sub-block is at least m (or $m + 1$ if this sub-block corresponds to the first subinterval). Therefore it is possible to extract $2m$ (or $2(m+1)$ if this sub-block corresponds to the first subinterval) linearly independent rows from the current sub-block and therefore the corresponding matrix is non-singular. □

Theorem 4.4.4. *If*

$$N_1 - m - 1 < K_1\,, \tag{4.34}$$

or

$$N_l - m < K_l,\; l = 2, \ldots n, \tag{4.35}$$

where N_l and K_l , $l = 1, \ldots n$ are the total number of recordings in l-th subinterval and the multiplicity coefficients for l-th subinterval respectively. Then, matrix $\boldsymbol{B}$ is rank-deficient.

Proof of Theorem 4.4.4. Similar to Theorem 4.4.3, for each diagonal block of matrix $\boldsymbol{A}$ the corresponding two diagonal sub-blocks can be obtained. If (4.34) or (4.35) holds, then, the second diagonal block of $\tilde{\tilde{\boldsymbol{A}}}_d$ contains at least one zero row. Therefore matrix $\boldsymbol{B}$ is singular. □

Thus, the only situation that is not covered at all is when

$$K_1 < m + 1 \text{ or } K_i < m,\; i = 2, \ldots, n.$$

The following example demonstrates that the gap can not be closed.

Example 4.4.1. *Assume that there are two subintervals and θ is the only internal knot. The degree of the corresponding polynomial pieces is two (quadratic, $m = 2$). Suppose that it is possible to extract three linearly independent rows for the first subinterval, while for the second interval $N_2 = 4$ (four recordings t_1, t_2, t_3 and t_4 at the second subinterval) $K_2 = 1$ (no repeated value for $g(t_i)$ at $t_i,\ i = 1, \ldots, 4$). Clearly, we can not apply Theorem 4.4.3 or Theorem 4.4.4. Assume for simplicity that*

$$(t_1 - \theta) = 1,\ (t_2 - \theta) = 2,\ (t_3 - \theta) = 3,\ (t_4 - \theta) = 4.$$

Now consider two situations (m, N_2 and K_2 are the same). In the first situation, the matrix is singular while in the second situation, it is full-rank.

1. $g(t_1) = 0,\ g(t_2) = 1,\ g(t_3) = 2,\ g(t_4) = 3.$ *In this case, the block that corresponds to the second subinterval is a square matrix*

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 2 & 4 & 2 & 4 \\ 3 & 9 & 6 & 18 \\ 4 & 16 & 12 & 48 \end{pmatrix},\ \det(\boldsymbol{A}) = 0.$$

2. $g(t_1) = 0,\ g(t_2) = 1,\ g(t_3) = -1,\ g(t_4) = 3.$ *In this case, the block that corresponds to the second subinterval is a square matrix*

$$\boldsymbol{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 2 & 4 & 2 & 4 \\ 3 & 9 & -3 & -9 \\ 4 & 16 & 12 & 48 \end{pmatrix},\ \det(\boldsymbol{A}) = -432.$$

Therefore the same combination of $N_i, K_i,\ i = 1, \ldots, n$ and m may lead to both singular and non-singular matrices, depending on the prototype function.

4.5 Application to Signal Processing

In this section, we give two examples (one for each model) of how our non-singularity verification conditions can be applied in practice. We demonstrate that the properties of g play the core role in singularity verification. In particular, a slight change in g changes the singularity of the system matrix. This issue has been partially demonstrated in Example 4.4.1. In this section, we provide another example where the singularity was verified using Theorems 4.4.3 and 4.4.4. In addition, we propose an algorithm for signal approximation where the choice of optimization techniques is based on the corresponding singularity study.

4.5.1 Model 1

An EEG (electroencephalogram, also known as brain wave) signal is modeled as a sine wave

$$W_1 = S(\mathbf{x}, \boldsymbol{\tau}, t_i)\sin(\omega t_i + \tau)\,, \tag{4.36}$$

where S is the spline function defined in (4.1) whose knots $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n-1})$ are equidistant. The frequency and phase shift are shown by ω and τ respectively. For each combination of ω and τ the corresponding optimization problem is

$$\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - W_1)^2\,. \tag{4.37}$$

This is an LLSP. To achieve the best combination of ω and τ we run a double loop over the defined intervals of ω and τ. Then, we keep the optimal combination that is the combination of ω and τ values with the lowest objective function value (see Section 4.5.3 for details). Optimization problem (4.37) is formulated as

$$\min_{\mathbf{x}} \|\boldsymbol{M}\mathbf{x} - \mathbf{y}\|_2^2\,, \tag{4.38}$$

where $\mathbf{y} = (y_1, y_2, \ldots, y_N) \in \mathbb{R}^N$ are the recorded signal values at $t_i \in \mathbb{R}^N$, $\mathbf{x} \in \mathbb{R}^{mn+1}$, and $\boldsymbol{M}$ is a matrix with N rows and $mn + 1$ columns. If the system matrix $\boldsymbol{M} \in \mathbb{R}^{N\times(mn+1)}$ is a full-rank matrix, then, this LLSP can be solved through the system of normal equations.

In this application, there are natural restrictions on ω and n that are considered in [57].

- Frequency ω is a parameter that is normally assigned by a manual scorer (medical doctor). Therefore the value for frequency is bounded from above (by $16\,\mathrm{Hz}$) and restricted to integer (due to human scorer's perception limitations).
- The duration of the events is between 0.5 and 3 seconds (these events are called K-complexes) [57]. It is not reasonable to consider any subinterval shorter than 1 second. Since the duration of the original signal is 10 seconds, the number of subintervals n can not exceed 10.

We need to show that there are $mn+1$ linearly independent rows to prove that $\boldsymbol{M}$ is a full-rank matrix (see Theorem 4.4.1). Before we start proving that $\boldsymbol{M}$ is a full-rank matrix, we have to estimate the number of constants $\sin(\omega t_i+\tau)$ being zero (estimation of $K_l, l=1,\dots,n$) where τ is a phase shift. It can happen no more than $2\omega D+1$ times for a D seconds duration of an EEG. Suppose that the knots $\theta_1,\dots,\theta_n$ are equidistant. According to the reported experiments [100], $N=1000$ (total number of recordings), $n=5$ (the number of subintervals), $m=4$ (degree of spline pieces) and ω did not exceed $16\,\mathrm{Hz}$ and confined to integers due to human scoring limitations. Since the duration of each signal segment is 10 seconds, the duration of each subinterval is

$$D/n = 2 \text{ seconds}\,.$$

$N_l=200$ is the total number of recordings in $l-$th subinterval and K_l is the number of time moments t_i in the $l-$th subinterval when $g(t_i)=0$. Due to the periodicity of sine function

$$K_l \le 2\omega \times D/n + 1 = 65\,,\; l=1,\dots,n\,,$$

and therefore

$$N_l - K_l = 200 - 65 \ge m = 4,\; l=1,\dots,n.$$

Hence, due to Theorem 4.4.1, matrix $\boldsymbol{M}$ is non-singular. So, the normal equations method can be applied.

4.5.2 Model 2

Similar to Model 1, the knots are equidistant, $N = 1000$, $n = 5$, $m = 4$ and ω did not exceed 16 Hz. An EEG signal is modeled as a sine wave. It is shifted vertically (signal biasing) by a spline function as

$$W_2 = S_1(\mathbf{x}_1\,, \boldsymbol{\tau}\,, t_i)\sin(\omega t_i + \tau) + S_2(\mathbf{x}_2\,, \boldsymbol{\tau}\,, t_i)\,. \tag{4.39}$$

Therefore the corresponding optimization problem is

$$\min_{\mathbf{x}} \sum_{i=1}^{N} (y_i - W_2)^2\,, \tag{4.40}$$

where $\mathbf{x} = [\mathbf{x}_1; \mathbf{x}_2] \in R^{2mn+2}$ and $y_i \in \mathbb{R}^N$ are the recorded signal values at t_i for $i = 1, 2, \ldots, N$. The dimension of this problem is $2mn + 2$. The optimization problem (4.40) can be rewritten as

$$\min_{\mathbf{x}} ||\boldsymbol{B}\mathbf{x} - \mathbf{y}||_2^2\,, \tag{4.41}$$

where $\boldsymbol{B} \in \mathbb{R}^{N\times(2mn+2)}$ is described in Section 4.3.2, $\mathbf{x} \in \mathbb{R}^{2mn+2}$ and $\mathbf{y} \in \mathbb{R}^N$.

Example 4.5.1. *A* 10 *second segment of data (original signal) is to be approximated (Model 2). The approximation parameters are* $N = 1000$*,* $n = 5$*,* $m = 4$*,* $t_1 = 0$*,* $\tau = 0$ *and* $\omega = 50$ *Hz. The knots are equidistant,* $g(t) = \sin(t)$ *and the frequency of recording is* 100 *Hz (*100 *recordings per second). Thus,* $g(t_i) = 0,\ i = 1, \ldots, 1000.$ *By applying Theorem 4.4.4, one can conclude that matrix* $\boldsymbol{B}$ *is rank-deficient. Now consider slightly different settings. All the parameters remain the same except* τ*, which is now* $\pi/2$*. If* i *is odd* $g(t_i) = 1$*. Otherwise* $g(t_i) = -1$ $(i = 1, \ldots, 1000)$*. In this case, by applying Theorem 4.4.3, one can conclude that the system matrix is full-rank. Note that similar conclusion can be made for any* $\tau \neq k\pi,\ k \in \mathbb{Z}$*.*

In many practical situations, the conditions of Theorems 4.4.3-4.4.4 are not satisfied and therefore the singularity is not verified. However, when m, n, ω and recording frequencies are increasing, the determinants of the corresponding matrices in normal equations become very close to zero. This should be also taken into account in the numerical experiments.

To analyze the non-singularity of matrix $\boldsymbol{M}$ in Model 1, it is sufficient to have enough time moments t_i where the corresponding values $g(t_i) \neq 0$, while in Model 2, it is not important which values of $g(t_i)$ are taking, but how many of them give the same value. This adds additional difficulty in the singularity analysis of Model 2.

Remark. Singularity verification rules for Models 1 and 2 involving polynomials as the amplitude are as follows.

- A necessary and sufficient condition for non-singularity of Model 1 can be developed by assigning $n = 1$ to the conditions of Theorem 4.4.1.
- Sufficient conditions for non-singularity and singularity of Model 2 can be developed by assigning $n = 1$ to the conditions of Theorems 4.4.2, 4.4.3 and 4.4.4.

The definition of a polynomial is presented in Equation (3.2) of Chapter 3.

4.5.3 Algorithm Implementation

In this section, we present an algorithm for solving (4.5) and (4.6). In most practical problems, ω and τ are not known in advance and therefore there should be a procedure for choosing them. One way is to consider them as additional decision variables. This approach is not very efficient since the corresponding optimization problems become non-convex and can not be solved efficiently [57]. Therefore we can assign exact values from defined intervals of ω and τ that form a fine grid (using double loops) instead of optimizing them directly. Then, we solve the corresponding LLSPs and keep the best obtained results [100].

Algorithm 4.2 can be used to solve a sequence of LLSPs. In this algorithm, ω_0 and ω_f are the initial and final values for ω. Similarly, τ_0 and τ_f are the initial and final values for τ.

Remark.

1. In most our experiments $\omega_0 = 1\,\text{Hz}$, $\omega_f = 16\,\text{Hz}$, $\tau_0 = 0$ and $\tau_f = \pi$. These parameters may vary for different applications.

2. Theorem 4.4.1 (Model 1) provides necessary and sufficient non-singularity conditions, while Theorem 4.4.3 (Model 2) only provides sufficient non-singularity conditions but not necessary. Therefore it may be possible that the conditions of Theorem 4.4.3 are not satisfied, nevertheless the corresponding system matrices are full-rank. However, since we can not confirm non-singularity, it is safer to use SVD or QR decomposition.

Algorithm 4.2 Signal approximation through LLSPs

```
Specify the initial and final values for the frequency ($\omega_0$ and $\omega_f$) and phase shift ($\tau_0$ and $\tau_f$).
for $\omega = \omega_0 : \omega_f$ do
    for $\tau = \tau_0 : \tau_f$ do
        set an LLSP with fixed $\omega$ and $\tau$
        if the conditions of Theorem 4.4.1 (Model 1) or Theorem 4.4.3 (Model 2) are satisfied then solve the problem through Normal equations
        else solve the LLSP through SVD or QR decomposition
        end if
        record the optimal values of the objective function, $\omega$, $\tau$ and $\mathbf{x}$
    end for
end for
```

4.6 Uniform Approximation-based Models

Models 3 and 4 are considered such that Model 3 corresponds to the case when the wave is oscillating around “zero level” while Model 4 enables a vertical shift (signal biasing) in a form of a polynomial spline. A signal segment $\mathbf{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$ is considered where $y_i,\ i = 1, \ldots, N$ are evaluated at time $t_i,\ i = 1, \ldots, N$. This signal is approximated by a function $f(t)$ defined in Equations (4.3) and (4.4) to form Models 3 and 4 respectively. These two approximation models are formulated as follows:

$$\text{Model 3}: \min_{\mathbf{x}} \quad \max_{i=1,\ldots,N} |y_i - S(\mathbf{x}, t_i)g(t_i)|, \tag{4.42}$$

$$\text{Model 4}: \min_{\mathbf{x_1},\mathbf{x_2}} \quad \max_{i=1,\ldots,N} |(y_i - S_1(\mathbf{x}_1, t_i)g(t_i) - S_2(\mathbf{x}_2, t_i)|, \tag{4.43}$$

where S, S_1, S_2 and g are described in Section 4.3.

Models 3 and 4 are non-smooth convex optimization problems based on the uniform (Chebyshev) approximation. There exist many methods to solve non-smooth convex problems for instance, subgradient-based method and smoothing approximation methods [34]. These methods have some drawbacks as follows.

- The subgradient algorithms are very slow for solving practical problems.
- The nature of EEG signals used in this study is non-smooth because of the sudden changes in the signal amplitude. The smoothing approximations of signals do not enable one to capture the abrupt changes in the amplitude. So, the smoothing approximation method may not be a suitable representative of an EEG signal. Hence, it is not accurate and efficient in this particular application.

Further, a few methods have been proposed and reported in the literature dealing with the development of optimality conditions for the uniform (Chebyshev) approximation appearing in Models 3 and 4 recently. Sukhorukova and Ugon in 2016 [85] obtain optimality conditions for the uniform (Chebyshev) approximation where a modeling function is approximated by linear combinations of fixed knots splines with weighting functions. In particular, by assigning a sine function as the weighting function, one can obtain optimization problems appearing in Model 3 while optimality conditions for Model 4 remain an open problem.

One possible way to verify that the solutions to Models 3 and 4 are optimal is via a linear programming reformulation of aforementioned models [81]. Once this is done they can be solved using the existent techniques such as an interior point method efficiently and reliably.

4.6.1 Model 3

Suppose that the degree of a spline is m, the number of subintervals is n and the spline knots are defined in Equation (4.7). It is reformulated as a linear programming problem (LPP). Let

$$f(\mathbf{x}) = \max_{t_i,\ i=1,\dots,N} \left| y_i - S(\mathbf{x}, t_i) g(t_i) \right|, \tag{4.44}$$

then, $f(\mathbf{x})$ is a convex function but the maximum of absolute values is not a linear function. A new variable Z is introduced to make $f(\mathbf{x})$ as a linear function. Assume that

$$Z = \max_{t_i,\ i=1,\ldots,N} |y_i - S(\mathbf{x}, t_i)g(t_i)|\,, \tag{4.45}$$

therefore the new model is

$$\min_{\mathbf{x}} Z \quad \text{subject to} \quad Z = \max_{t_i,\ i=1,\ldots,N} |y_i - S(\mathbf{x}, t_i)g(t_i)|\,, \tag{4.46}$$

now, there exist a non-linear function in the constraint however, the following equivalent reformulation takes care of this issue.

$$\min_{\mathbf{x}} Z \quad \text{subject to} \begin{cases} [S(t_i)g(t_i)]\,\mathbf{x} \le y_i + Z\boldsymbol{e}\,, \\ -[S(t_i)g(t_i)]\,\mathbf{x} \le -y_i + Z\boldsymbol{e}\,, \end{cases} \tag{4.47}$$

where $\boldsymbol{e} \in \mathbb{R}^{N\times 1}$ is a matrix of ones or unit matrix. Since Z is a variable, the LPP defined in (4.47) is rewritten as follows:

$$\begin{bmatrix} \boldsymbol{M} & -\boldsymbol{e} \\ -\boldsymbol{M} & -\boldsymbol{e} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ Z \end{bmatrix} \le \begin{bmatrix} y_i \\ -y_i \end{bmatrix}, \tag{4.48}$$

where $\boldsymbol{M}$ and $-\boldsymbol{M}$ have N rows of the form $S(t_i)g(t_i)$ and $-S(t_i)g(t_i)$ respectively with $mn+1$ columns, $\mathbf{x} \in \mathbb{R}^{mn+1}$ and $y_i \in \mathbb{R}^N$. The solutions to the LPP defined in (4.47) can be obtained through solving the system of

$$\boldsymbol{Q}\boldsymbol{X} \le \mathbf{Y}\,. \tag{4.49}$$

$\boldsymbol{Q} \in \mathbb{R}^{2N\times(mn+2)}$ is a block matrix containing submatrices of $\boldsymbol{M} \in \mathbb{R}^{N\times(mn+1)}$, $-\boldsymbol{M} \in \mathbb{R}^{N\times(mn+1)}$ and $-\boldsymbol{e} \in \mathbb{R}^{N\times 1}$. $\mathbf{X}$ contains a vector of $\mathbf{x}$ and a variable Z. The y_i and $-y_i$ are included in vector $\mathbf{Y}$.

An LPP's solution that is optimal can be obtained by solving the linear system defined in Equation (4.49).

4.6.2 Model 4

In this model, the wave defined in Equation (4.4) is used to approximate an EEG signal. Similar to Model 3, the spline degree is m, the number of subinterval is n and the corresponding knots are

$$t_0 = \theta_0 \leq \theta_1 \leq \theta_2 \leq \cdots \leq \theta_{n-1} \leq \theta_n = t_N \,.$$

A linear programing formulation of Model 4 is

$$\min_{\mathbf{x_1},\mathbf{x_2}} Z \quad \text{subject to} \begin{cases} [S_1(t_i)g(t_i) + S_2(t_i)]\,[\mathbf{x}_1;\mathbf{x}_2] \leq y_i + Z\boldsymbol{e}\,, \\ -[S_1(t_i)g(t_i) + S_2(t_i)]\,[\mathbf{x}_1;\mathbf{x}_2] \leq -y_i + Z\boldsymbol{e}\,, \end{cases} \tag{4.50}$$

where S_1 and S_2 are polynomial splines with fixed knots, $g(t)$ is a prototype function, $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{mn+1}$, $\boldsymbol{e} \in \mathbb{R}^{N\times 1}$ is an unit matrix and

$$Z = \max_{t_i,\; i=1,\ldots,N} |y_i - S_1(\mathbf{x}_1, t_i)g(t_i) - S_2(\mathbf{x}_2, t_i)|\,. \tag{4.51}$$

Let $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2] \in \mathbb{R}^{2mn+2}$ then, the LPP defined in (4.50) can be rewritten as follows:

$$\begin{bmatrix} \boldsymbol{B} & -\boldsymbol{e} \\ -\boldsymbol{B} & -\boldsymbol{e} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ Z \end{bmatrix} \leq \begin{bmatrix} y_i \\ -y_i \end{bmatrix}, \tag{4.52}$$

where matrix $\boldsymbol{B}$ has N rows of the form $S_1(t_i)g(t_i) + S_2(t_i)$ and $2mn + 2$ columns, $\mathbf{x} \in \mathbb{R}^{2mn+2}$ and $y_i \in \mathbb{R}^N$. The optimal solution of an LPP defined in (4.50) is found through solving the following system.

$$\boldsymbol{E}\boldsymbol{X} \leq \mathbf{Y}\,. \tag{4.53}$$

$\boldsymbol{E} \in \mathbb{R}^{2N\times(2mn+3)}$ is a block matrix containing submatrices of $\boldsymbol{B} \in \mathbb{R}^{N\times(2mn+2)}$, $-\boldsymbol{B} \in \mathbb{R}^{N\times(2mn+2)}$ and $-\boldsymbol{e} \in \mathbb{R}^{N\times 1}$. $\mathbf{X}$ contains a vector of $\mathbf{x} \in \mathbb{R}^{2mn+2}$ and a variable Z. The $y_i \in \mathbb{R}^N$ and $-y_i$ are included in vector $\mathbf{Y}$.

Remark. The linear programming reformulation of Models 3 and 4 where a polynomial P is approximated as an amplitude is similar to ones whose amplitudes are approximated by splines S.

4.7 Summary

This chapter presents the singularity study of LLSPs (Model 1 and Model 2) involving fixed knots polynomial splines and polynomials in order to find better suited methods to solve Models 1 and 2 and enhance the efficiency of the approximation algorithms. This issue is especially important when one needs to solve the corresponding problems repeatedly. This is the case for an approximation algorithm where the experiments are run by assigning different values to ω and τ rather than optimizing them.

Due to the corresponding optimization problem complexity ω and τ are not optimized. Most LLSPs can be solved through the system of normal equations if the system matrix is non-singular otherwise they can be solved through a more robust approach like an SVD. Further, linear programming reformulations of non-smooth convex optimization problems (Model 3 and Model 4) are presented and their solutions are optimal.

To illustrate and evaluate the efficiency of four convex optimization models numerical simulations are run in Chapters 5 and 6. Those models are used as feature extraction methods to detect transient events in EEG signals.

Chapter 5

Medium Scale Dataset of an EEG signal for K-complex Detection

5.1 Introduction

In the previous chapters the structure of developed convex optimization models as optimization-based preprocessing approaches (OPAs) was discussed. In this chapter, the efficiency and effectiveness of OPAs are illustrated and evaluated through numerical simulations. Their performance to extract the essential features of an EEG signal is verified in numerical simulations of an automated detection of K-complexes problem. The OPAs are used as preprocessing approaches for extracting the key features of an EEG signal in presence of K-complexes since it is appeared that the automated detection of such transient event is already difficult enough. This automation is difficult due to the ambiguity of the scoring rules, rough description of a K-complex and complexity of dataset. To investigate the improvements in the classification accuracy that can be achieved through employing OPAs the classification algorithms (classifiers) are applied on the original dataset (raw dataset) and a dataset after OPAs. The efficiency of an OPA to extract the key features of an EEG signal significantly affect the classification accuracy obtained from classifiers. Therefore this study is confined to the development of feature extraction methods but not classifiers.

5.2 The Performance of OPAs

Within the feature extraction stage of the classification problem, four convex optimization models denoted by LLSOM1, LLSOM2, UOM1 and UOM2 are developed. The first two models are based on a sequence of linear least squares problems and the last two ones are based on the uniform (Chebyshev) approximation. The normal equations method is used to solve the first model since $\boldsymbol{M}$ is a full-rank matrix. LLSOM2 is solved using an SVD method because $\boldsymbol{B}$ is a rank-deficient matrix. The last two models result in convex optimization problems and are solved using the CVX [22, 33], and linear programming routine in MATLAB.

The signal amplitude is approximated by the polynomial (P) and spline (S) functions. They are described by Equations (3.2) and (3.4) in Chapter 3, respectively. Let m_1 and m_2 be the degrees of a spline and polynomial functions respectively. According to the technical specification of an EEG K-complex dataset provided in Section 3.10.1 of Chapter 3, $m_1 = 4$ (degree of a spline) and $n = 5$ (the number of subintervals) are assigned to the spline function S. The knots are chosen as a sequence of equidistant knots. The frequency grid is specified as the numbers between $\omega_0 = 0.1\,\text{Hz}$ and $\omega_f = 14.1\,\text{Hz}$ with the step size of $1\,\text{Hz}$. Note that the medical practitioners almost never consider frequencies above $16\,\text{Hz}$. The interval $[0, \pi]$ with the step size $\pi/4$ was assigned to τ. In order to balance the number of parameters in a spline and a polynomial functions, the degree of P is increased to $m_2 = m_1 n$ where $m_1 = 4$ and $n = 5$. So, $m_2 = 20$ is assigned to P. Herein, the results from four models involving fixed knots splines S are comparable with ones involving polynomials P. Three additional parameters that characterize the improvement of the objective function after optimization-based preprocessing are considered as essential features. These three parameters are the value of objective function, ω and τ.

In LLSOM1 and UOM1 involving fixed knots splines (S), the output dimension is $m_1 n + 1$ (21 where $m_1 = 4$ and $n = 5$) while the output dimension of ones involving polynomials P is $m_2 + 1$ (21 where $m_2 = 20$). Therefore $N = 1000$ features of the original dataset have been reduced to $24 = m_1 n + 4 = m_2 + 4$ essential features. The dimension of essential features is illustrated in Figure 5.1.

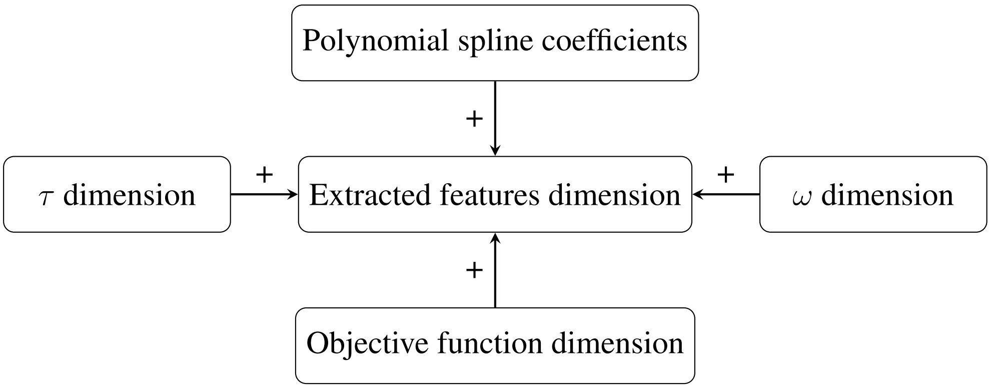
Polynomial spline coefficients
+
τ dimension
+
Extracted features dimension
+
ω dimension
+
Objective function dimension

Figure 5.1: The dimension of extracted features after OPAs

For the optimization models LLSOM2 and UOM2 involving the spline function S, the output dimension is $2m_1n + 2$ (42 where $m_1 = 4$ and $n = 5$) whereas the output dimension of ones involving polynomials P is $2m_2 + 2$ (42 where $m_2 = 20$). By considering three additional parameters of objective function, ω and τ, $N = 1000$ features of the original dataset have been reduced to $2m_1n + 5 = 2m_2 + 5 = 45$ essential features. Figure 5.1 depicts the dimension of extracted features.

5.3 The Numerical Results of Feature Extraction Methods

In this section, the results obtained from four convex optimization models called LLSOM1, LLSOM2, UOM1 and UOM2 that are also referred as OPAs are presented in Tables 5.1 to 5.4. These tables give the CPU time in seconds, mean frequency of segments containing K-complexes and non-K-complexes, and mean of objective function values for each model. The EEG signal and its approximation for each model are illustrated in Figures 5.2 to 5.9. In the feature extraction stage of classification problems, OPAs are applied over the EEG recordings in presence of K-complexes and detect such transient event automatically. The OPAs enable one to

- extract the key features of original signal such that obtained features are describing the properties of signal accurately,
- reduce the size of dataset,
- enhance the performance of classifiers.

In this part, the numerical behaviors of each OPA are analyzed.

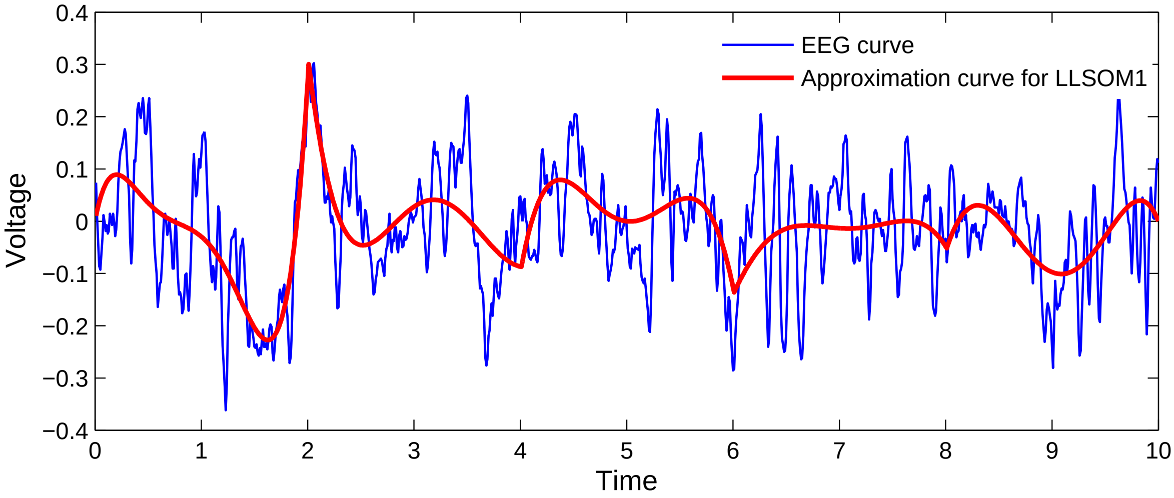

Figure 5.2: Approximation curve after LLSOM1-based preprocessing involving spline S.

5.3.1 LLOSM1

The first model is a sequence of LLSPs when $\omega = \omega_0 : 1 : \omega_f$ and $\tau = \tau_0 : \pi/4 : \tau_f$ are constants. Nevertheless, $\omega_0 = 0.1\,\text{Hz}$, $\omega_f = 14.1\,\text{Hz}$ and $\tau_0 = 0$, $\tau_f = \pi$ are considered according to the specification of proposed dataset. This model is described in detail in Section 3.5.1 of Chapter 3.

The spline function (S) of degree $m_1 = 4$ whose $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{n-1})$ are fixed (equidistant) with $n = 5$ as the number of subintervals and the polynomial function (P) of degree $m_2 = 20$ with no interval divisions are used to approximate the amplitude. The normal equations method is utilized to solve this model since the system matrix $\boldsymbol{M}$ is full-rank. The formal proof is given in Chapter 4.

Figure 5.2 illustrates the approximation curve (red) and the original EEG signal (blue). The approximation amplitude S is considerably larger where the K-complex is located (between $2\,\text{sec}$ and $3\,\text{sec}$ of time segment). Although the approximation does not follow precisely the trend of original data, it is sufficient to detect the K-complex and therefore to produce the correct classification results.

Figure 5.3 illustrates the approximation curve in red and the original EEG signal in blue when the amplitude is approximated as a polynomial function P. The approximation amplitude P is almost larger where the K-complex is located between $2\,\mathrm{sec}$ and $3\,\mathrm{sec}$ of time segment. The degree of polynomial is increased to make its parameters comparable with the spline (S) parameters. Since the polynomial function of higher degree is used, the signal approximation problem becomes unstable and therefore the system matrix $\boldsymbol{M}$ may be close to singular. Therefore the polynomial function is not as flexible as the spline one to capture the sudden changes in amplitude. Table 5.1 demonstrates the numerical results obtained from LLSOM1 (optimization only, without classification). CPU time corresponds to the total preprocessing time (for the whole dataset). One can see that the CPU time for LLSOM1 involving polynomial function is higher than one for LLSOM1 involving spline function. It can be seen that $N = 1000$ features are reduced to 24 ones after LLSOM1-based preprocessing.

In addition, Table 5.1 indicates that the mean frequencies are significantly higher for non-K-complexes. This observation can be used for classification. Mean frequencies for K-complexes enable one to detect EEG waves containing them. They indicate that the frequency is an essential feature to detect K-complexes.

Table 5.1: Numerical results after LLSOM1-based preprocessing.

Numerical results	LLSOM1	
	Spline S	Polynomial P
Number of recordings (features)	1000	1000
CPU time (in seconds)	21	34
Output dimension [a]	24	24
MFK [b]	1.1000	1.1000
MFNK [c]	1.6385	1.7410
MOF [d]	2.9189	3.1019

[a] Number of extracted features.
[b] Mean Frequency for K-complexes (all instances).
[c] Mean Frequency for non-K-complexes (all instances).
[d] Mean of Objective Function values for whole segments.

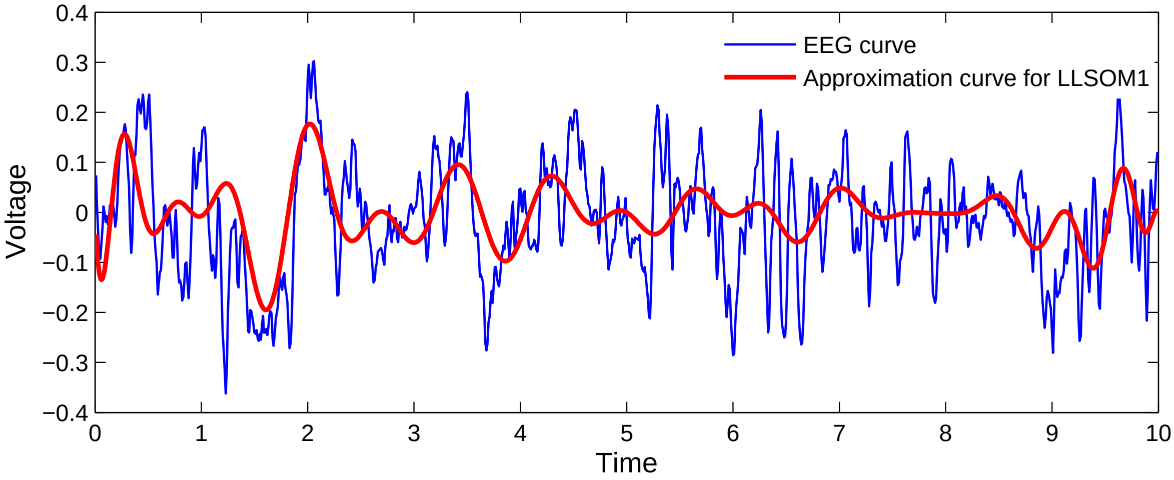

Figure 5.3: Approximation curve after LLSOM1-based preprocessing involving polynomial P.

5.3.2 LLOSM2

Similar to LLSOM1, LLSOM2 is a sequence of LLSPs when ω and τ are constants. The proposed intervals assigned to ω and τ are same as ones in LLSOM1. In this model, the wave defined in LLSOM1 is shifted vertically by a spline and a polynomial functions respectively.

$m_1 = 4$ and $m_2 = 20$ are the degrees of a spline and a polynomial respectively. The number of subintervals is $n = 5$. This model is detailed in Section 3.5.2 of Chapter 3. Matrix $\boldsymbol{B}$ contains $2m_2n+2$ and $2m_2+2$ columns when the amplitude is approximated by a spline and a polynomial respectively. It has N rows as well. Since $\boldsymbol{B}$ is a rank-deficient matrix (it is discussed in Chapter 4) then, an SVD method is used to solve this problem.

Figure 5.4 reveals that the approximation curve in the second optimization model (LLSOM2) is more accurate than LLSOM1 that is shown in Figure 5.2 when they used the spline function as an amplitude. It can be seen that the corresponding optimization model can detect the K-complexes where the amplitude of the approximation curve is considerably larger. In addition, they are located at the time intervals of $2\,\mathrm{sec}$ to $3\,\mathrm{sec}$, $3\,\mathrm{sec}$ to $4\,\mathrm{sec}$ and $5\,\mathrm{sec}$ to $6\,\mathrm{sec}$.

Figure 5.5 shows the approximation curve in LLSOM2 with the polynomial function

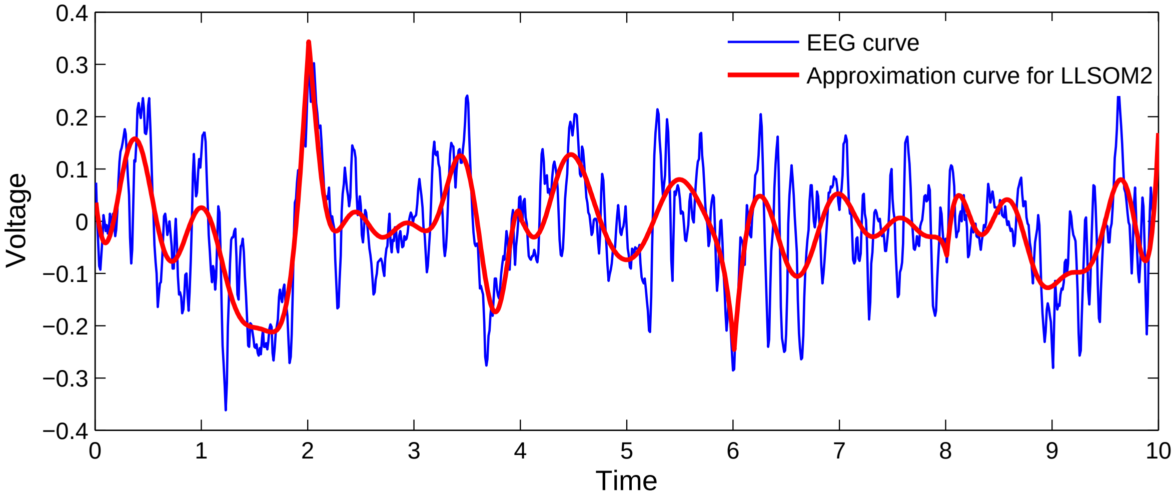

Figure 5.4: Approximation curve after LLSOM2-based preprocessing involving spline S.

is less accurate than with the spline one. As can be seen in Figure 5.5, LLSOM2 involving a polynomial function has trouble in detection of the K-complexes. Although this approximation does not follow the trend of the EEG curve, we proceed to produce the classification results.

It should be noted that the first few seconds of approximated signal produces a constant since the polynomial coefficients are extremely small and their absolute values are zero. There exists a trade off between a polynomial shape and a degree. In order to model the amplitude with a structure that is comparable with a spline function, the degree of a polynomial must be high. Therefore the number of parameters to be optimized is high. This can lead to highly unstable model. Polynomials of high degree are ill-famed for oscillations between exact fit values.

Table 5.2 depicts the numerical results for LLSOM2. The CPU times for LLSOM2 with a spline and a polynomial are close to each other. The objective function value (MOF) obtained from LLSOM2 with a spline is less than with a polynomial. Further, the value of MOF from LLSOM2 with a spline is less than from LLSOM1 in Table 5.1. The number of features ($N = 1000$) is reduced to 45.

Furthermore, Table 5.2 indicates that the mean frequencies are significantly higher for non-K-complexes. This observation can be used for the classification and shows that

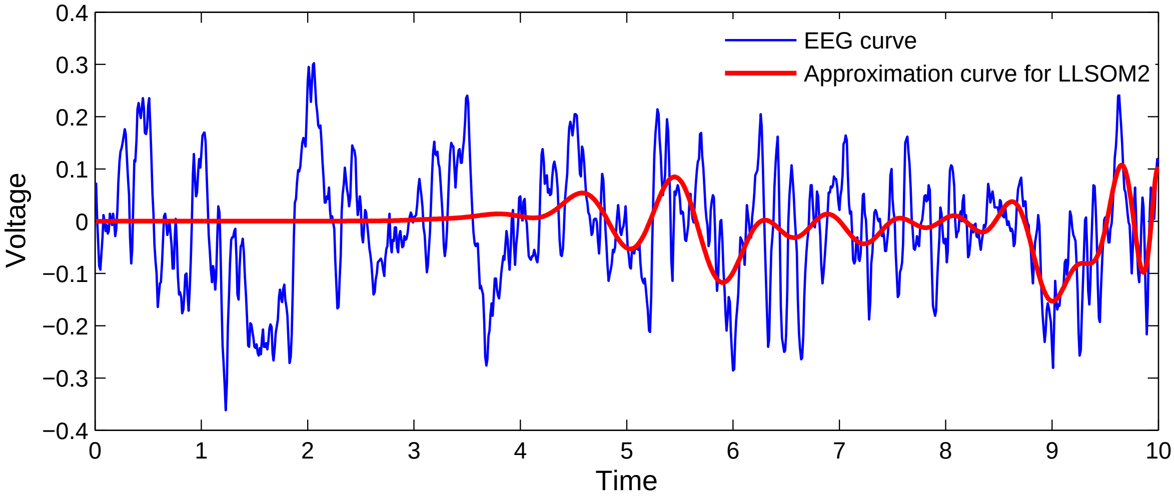

Figure 5.5: Approximation curve after LLSOM2-based preprocessing involving polynomial P.

the frequency is one of the key features to be extracted.

Table 5.2: Numerical results after LLSOM2-based preprocessing.

Numerical results	LLSOM2	
	Spline S	Polynomial P
Number of recordings (features)	1,000	1,000
CPU time (in seconds)	346	376
Output dimension	42	42
MFK	1.1000	1.3909
MFNK	1.7667	2.2282
MOF	2.5517	3.2419

5.3.3 UOM1

The third model is based on the uniform (Chebyshev) approximation. In UOM1, the wave is approximated in the same way as LLSOM1. This model minimizes the maximum of the absolute deviation between the original data y_i and the approximated wave. It is detailed in Section 3.5.3 of Chapter 3. First, $A(t_i)$ is approximated by a spline function (S) of degree $m_1 = 4$ with $n = 5$ as the number of subintervals. Second, it is approximated by a polynomial function (P) of degree $m_2 = 20$ with no interval divisions. This problem is programmed and solved by the CVX package and

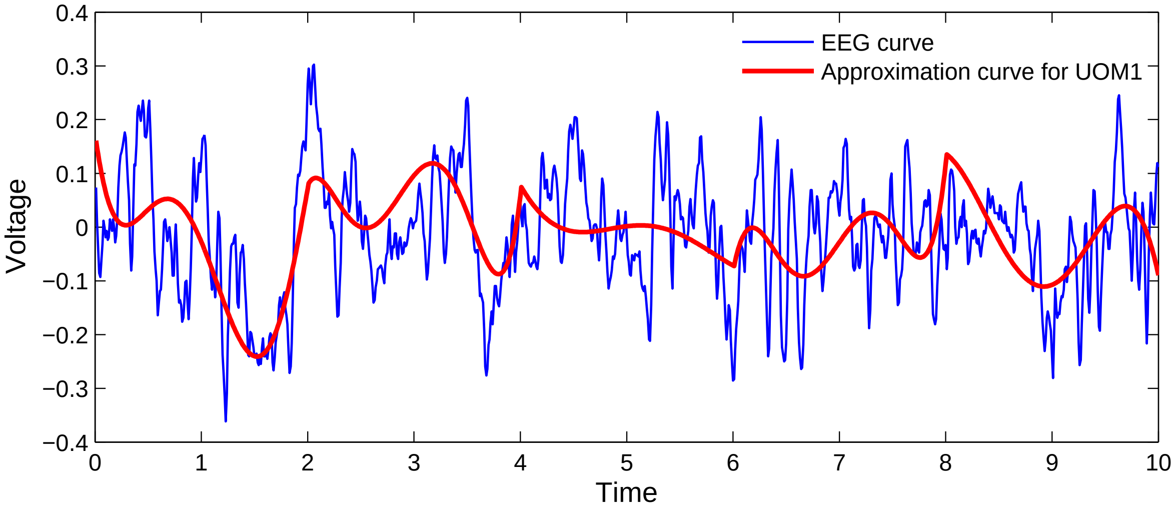

Figure 5.6: Approximation curve after UOM1-based preprocessing involving spline S.

the LINPROG subroutine in MATLAB. The $N = 1000$ features are reduced to 24 ones after UOM1-based preprocessing over EEG recordings.

The approximation curve obtained from UOM1 described in (3.18) of Chapter 3 is shown in Figure 5.6. The spline function is used to approximate the amplitude. Although this approximation does not follow the trend of the EEG curve, we proceed to produce the classification results.

In addition, the amplitude is approximated as a polynomial function of degree $m_2 = 20$. The high degree polynomial is chosen to be comparable with the structure of a spline. The numerical results indicate that both prime and dual problems are infeasible and different solvers in CVX fail during their implementation. As mentioned earlier, the high degree polynomial ($m_2 = 20$) approximated as an amplitude results in highly unstable model.

To evaluate the performance of a polynomial in approximation of an amplitude the degree of $m_2 = 5$ is selected. Note that theses results are not comparable with those ones obtained from UOM1 involving the spline function since the number of parameters for a spline is $m_1 n + 1 = 21$ whereas for a polynomial is $m_2 + 1 = 6$. Therefore the classifiers are not employed over the set of features obtained from UOM1 containing the polynomial of degree $m_2 = 5$.

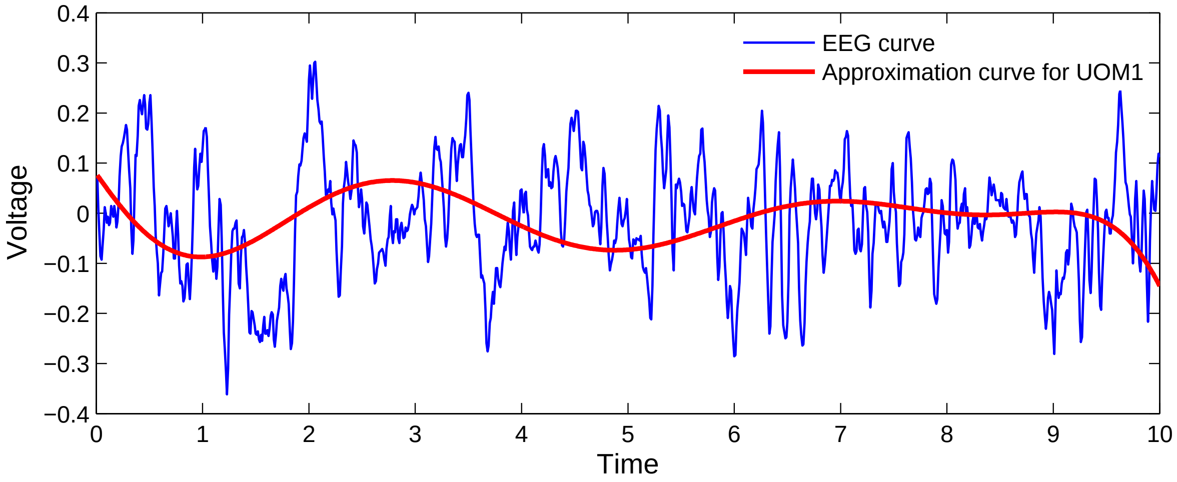

Figure 5.7: Approximation curve after UOM1-based preprocessing involving polynomial P.

Figure 5.7 illustrates the approximation curve obtained from UOM1 with polynomial function of degree $m_2 = 5$ as an amplitude. It can be seen that this feature extraction method fails to detect the K-complexes and this approximation dose not follow the trend of the EEG signal.

Table 5.3 demonstrates the numerical results for UOM1. Two solvers from CVX are used. The default one is SDPT3 and the professional one is GUROBI. The GUROBI has less computation time than LINPROG and SDPT3 for implementation of UOM1 containing the spline. The mean frequencies are significantly higher for non-K-complexes. This observation can be used for the classification. The MOF has less value than other MOFs obtained from LLSOM1 and LLSOM2 when a spline is approximated as an amplitude. The number of features ($N = 1000$) is reduced to 24 after UOM1-based preprocessing containing the spline.

Table 5.3: Numerical results after UOM1-based preprocessing.

Numerical results	CVX SDPT3		LINPROG		CVX GUROBI
	$S^\star$	$P^{\star\star}$	$S^\star$	$P^{\star\star}$	$S^\star$
Number of recordings	1,000	1,000	1,000	1,000	1,000
CPU time (in seconds)	2,654	1,341	2,300	1,387	1,534
Output dimension	21	6	21	6	21
MFK	1.1000	1.7452	1.1000	1.7452	1.1000
MFNK	2.2538	4.1513	2.2538	4.1513	2.2538
MOF	0.2922	0.3290	0.2922	0.3290	0.2922

* The degree of spline (S) is $m_1 = 4$ and $n = 5$ is the number of subintervals.
** The degree of polynomial (P) is $m_2 = 5$.

5.3.4 UOM2

Similar to UOM1, UOM2 is based on the uniform approximation. In UOM2, the EEG signal is approximated similarly to LLSOM2. This model is presented in detail in Section 3.5.4 of Chapter 3. There is no solution for this model when a polynomial of degree $m_2 = 20$ is approximated as the amplitude with the same reasoning as described for UOM1.

Figure 5.8 presents the approximation curve using UOM2 with the spline. In Figure 5.8, there is a clear trend of increasing the frequency of each segment rather than Figure 5.6. One can see that the amplitude of the approximation curve is considerably larger at the time intervals of $2\,\mathrm{sec}$ to $3\,\mathrm{sec}$ and $3\,\mathrm{sec}$ to $4\,\mathrm{sec}$ (possible K-complex).

Finally, Figure 5.9 illustrates the approximation curve using UOM2 with the polynomial of degree $m_2 = 5$. Therefore UOM2 involving the polynomial fails to detect K-complexes and it does not follow the trend of the EEG curve. In conclusion, the spline function is more flexible than the polynomial one to detect the abrupt changes in amplitude.

Table 5.4 reveals the numerical results obtained from UOM2. One can see that the CPU time for UOM2 through GUROBI is less than LINPROG and SDPT3. Table 5.4 indicates that the mean frequencies are significantly higher for non-K-complexes. This observation can be used for classification. The number of features is reduced to 45 after

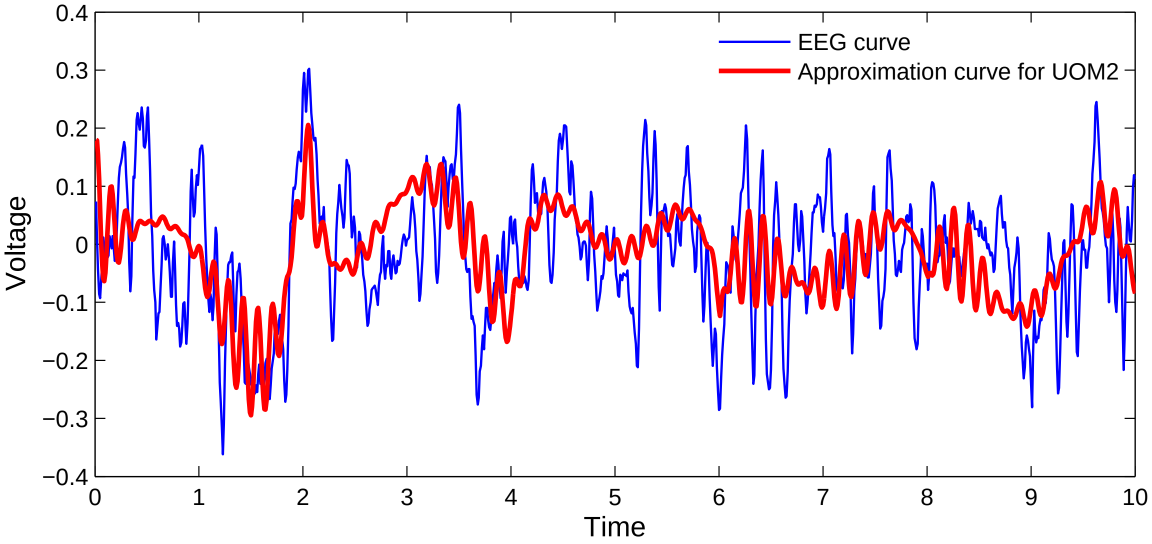

Figure 5.8: Approximation curve after UOM2-based preprocessing involving spline S.

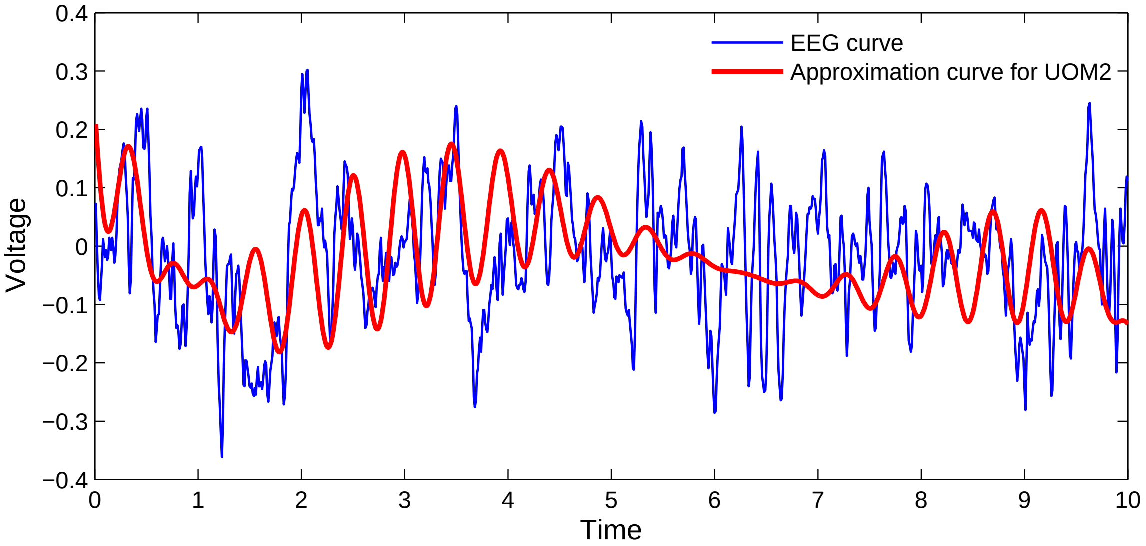

Figure 5.9: Approximation curve after UOM2-based preprocessing involving polynomial P.

Table 5.4: Numerical results after UOM2-based preprocessing.

Numerical results	CVX SDPT3		LINPROG		CVX GUROBI
	$S^\star$	$P^{\star\star}$	$S^\star$	$P^{\star\star}$	$S^\star$
Number of recordings	1,000	1,000	1,000	1,000	1000
CPU time (in seconds)	4,270	2,371	4,143	2,305	3,632
Output dimension	42	12	42	12	42
MFK	2.1323	1.9065	2.1323	1.9065	2.1323
MFNK	3.5615	3.9718	3.5615	3.9718	3.5615
MOF	0.2348	0.3023	0.2348	0.3023	0.2348

* The degree of spline (S) is $m_1 = 4$ and $n = 5$ is the number of subintervals.

** The degree of polynomial (P) is $m_2 = 5$.

UOM2-based preprocessing containing the spline.

Data from Tables 5.1 to 5.4 indicates the remarkable results.

- The first two models are faster than the last two ones.
- UOM1 and UOM2 are not accurate on large scale datasets.
- Splines are more flexible and efficient to detect the abrupt changes in the amplitude than polynomials for this particular application.
- A comparison of the MOFs obtained from four models (LLSOM1, LLSOM2, UOM1 and UOM2) reveals that whenever the complexity of a model increases, its MOF value decreases gradually.
- The last model containing the spline has the least value of MOF.

As shown in Figures 5.2 to 5.9, LLSOM1 and LLSOM2 involving the spline can detect the K-complexes where the amplitude of the approximation curve is larger. Therefore splines perform better than polynomials in approximation of an amplitude.

5.4 Classification Results and Discussion

In the feature extraction stage described in Section 5.3, the key features of an EEG signal are extracted through LLSOM1–2 and UOM1–2. These models significantly reduce the dimension of the problem. Therefore $N = 1000$ features of the original signal is reduced to 24 and 45 with 70 segments after applying LLSOM1, UOM1 and LLSOM2, UOM2 respectively.

There is no solution for the last two models when the high degree polynomial is approximated as the amplitude. Therefore the results obtained from UOM1–2 involving the spline are considered to produce the classification results.

In order to evaluate the performance of the OPAs in detection of K-complexes, different statistical measurements for instance, ACC, sensitivity, specificity, FPR, FNR and the area under ROC curve are required. These metrics are described in Section 3.9 of Chapter 3. A range of different classifiers used in [57] is applied over the set of extracted features obtained from OPAs. These classifiers are defined in Section 3.8 of Chapter 3.

Table 5.5: Classification accuracy (ACC) on the test set for (a) the original dataset, 1000 features and (b) the preprocessed dataset (after optimization-based preprocessing when the spline ($m_1 = 4, n = 5$) is approximated as the amplitude), 24 features for LLSOM1 and UOM1, 45 features for LLSOM2 and UOM2.

ACC on (a)	ACC on (b)				Classifiers
	LLSOM1	LLSOM2	UOM1	UOM2	
47%	47%	47%	47%	47%	LibSVM
N/A	74%	68%	53%	63%	Logistic
74%	63%	74%	68%	53%	RBF Network
47%	63%	84%	47%	53%	SMO
53%	63%	74%	53%	79%	LazyIB1
53%	74%	79%	47%	69%	LazyIB5
47%	74%	74%	47%	68%	KStar
47%	74%	74%	47%	79%	LWL
37%	47%	47%	47%	47%	OneR
47%	74%	79%	47%	74%	J48
47%	74%	79%	47%	74%	J48graft
42%	74%	74%	53%	79%	LMT

First, the classifiers are used over the original dataset with $N = 1000$ features. The results are presented in the first column of Tables 5.5 and 5.6. It is apparent that "RBF Network" produces a good classification accuracy on the original dataset. The "Logistic" algorithm does not produce any result. This is most probably due to the memory limitations of the used software implementation. Second, all classifiers are

Table 5.6: Classification accuracy (ACC) on the test set for (a) the original dataset, 1000 features and (b) the preprocessed dataset (after LLSOM1 and LLSOM2 when the high degree polynomial ($m_2 = 20$) is approximated as the amplitude), 24 and 45 features for LLSOM1 and LLSOM2 respectively.

ACC on (a)	ACC on (b)		Classifiers
	LLSOM1	LLSOM2	
47%	47%	47%	LibSVM
N/A	47%	58%	Logistic
74%	63%	68%	RBF Network
47%	58%	53%	SMO
53%	63%	53%	LazyIB1
53%	53%	58%	LazyIB5
47%	58%	53%	KStar
47%	63%	58%	LWL
37%	53%	47%	OneR
47%	63%	58%	J48
47%	58%	58%	J48graft
42%	63%	58%	LMT

applied to the obtained set of features after optimization-based preprocessing. The classification accuracy results are presented in Tables 5.5 and 5.6 when a spline and a polynomial of degree $m_2 = 20$ are approximated as the amplitude respectively. Their corresponding confusion matrices (CMs) are presented in Tables 5.7 and 5.8.

Table 5.5 demonstrates the classification accuracy (ACC) on the original dataset and the set of features obtained from OPAs when the spline ($m = 4, n = 5$) is approximated as the amplitude. The accuracy of all classifiers except LibSVM and RBF Network is considerably improved by using the preprocessed dataset rather than the original dataset. The RBF Network classifier provides a better accuracy on the original dataset and LLSOM2 than LLSOM1, UOM1 and UOM2 and no classification method failed on the preprocessed dataset. In general, it is seen that faster optimization methods of LLSOM1 and LLSOM2 work quite well compared to the slow optimization methods of UOM1 and UOM2. Note that some of the classifiers perform better with specific optimization methods.

- Logistic works well after LLSOM1-based preprocessing;

- RBF Network, SMO, LazyIB5, J48 and J48graft perform well after LLSOM2-based preprocessing;
- LazyIB1, LWL and LMT produce better results after UOM2-based preprocessing;
- KStar works well after LLSOM1 and LLSOM2-based preprocessing.

In summary, all the classifiers except LibSVM and OneR are prominent in combination with LLSOM1–2 and UOM2.

Table 5.6 provides the ACC on the original dataset and the set of extracted features from LLSOM1–2 when the high degree polynomial ($m_2 = 20$) is approximated as the amplitude. Data from this table can be compared with the second and third columns of Table 5.5. This comparison shows that all the above classifiers are performed better after LLSOM1–2 when the spline is approximated as the amplitude.

The structure of a confusion matrix is described in Table 3.1 of Chapter 3. In the confusion matrices presented here, entry $\{11\}$ corresponds to the number of non-K-complexes classified correctly, entry $\{12\}$ corresponds to the number of non-K-complexes classified as K-complexes (the number of false positives), entry $\{21\}$ corresponds to the number of K-complexes classified as non-K-complexes (the number of false negatives) and entry $\{22\}$ corresponds to the number of K-complexes classified correctly.

One of the main goals is to develop an approach for automatic detection of K-complexes that is fast and accurate. To address this the number of false negatives has to be decreased. For further investigation, it is much better to highlight the suspect segments of K-complexes for doctors to accept or reject rather than eliminate them thoroughly.

As Shown in Table 5.7, the rate of false negatives of all classifiers except LibSVM, RBF Network and OneR is considerably decreased when the high degree polynomial is approximated as the amplitude.

The comparison between the Table 5.8 and the first column of Table 5.7 shows that the number of false negatives of all classifiers except for LibSVM, RBF Network and

Table 5.7: Confusion matrices (CMs) on the test set for (a) the original dataset, and (b) the preprocessed dataset (after LLSOM1 and LLSOM2 when the high degree polynomial ($m_2 = 20$) is approximated as the amplitude).

	CM on (b)		
CM on (a)	LLSOM1	LLSOM2	Classifiers
$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	LibSVM
N/A	$\begin{pmatrix} 4 & 5 \\ 5 & 5 \end{pmatrix}$	$\begin{pmatrix} 6 & 3 \\ 5 & 5 \end{pmatrix}$	Logistic
$\begin{pmatrix} 4 & 5 \\ 0 & 10 \end{pmatrix}$	$\begin{pmatrix} 2 & 7 \\ 0 & 10 \end{pmatrix}$	$\begin{pmatrix} 3 & 6 \\ 0 & 10 \end{pmatrix}$	RBF Network
$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 7 & 2 \\ 6 & 4 \end{pmatrix}$	$\begin{pmatrix} 8 & 1 \\ 8 & 2 \end{pmatrix}$	SMO
$\begin{pmatrix} 8 & 1 \\ 8 & 2 \end{pmatrix}$	$\begin{pmatrix} 7 & 2 \\ 5 & 5 \end{pmatrix}$	$\begin{pmatrix} 5 & 4 \\ 5 & 5 \end{pmatrix}$	LazyIB1
$\begin{pmatrix} 9 & 0 \\ 9 & 1 \end{pmatrix}$	$\begin{pmatrix} 4 & 5 \\ 4 & 6 \end{pmatrix}$	$\begin{pmatrix} 6 & 3 \\ 5 & 5 \end{pmatrix}$	LazyIB5
$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 3 & 6 \\ 2 & 8 \end{pmatrix}$	$\begin{pmatrix} 5 & 4 \\ 5 & 5 \end{pmatrix}$	KStar
$\begin{pmatrix} 8 & 1 \\ 9 & 1 \end{pmatrix}$	$\begin{pmatrix} 3 & 6 \\ 1 & 9 \end{pmatrix}$	$\begin{pmatrix} 4 & 5 \\ 3 & 7 \end{pmatrix}$	LWL
$\begin{pmatrix} 7 & 2 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 9 & 0 \\ 9 & 1 \end{pmatrix}$	$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	OneR
$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 3 & 6 \\ 1 & 9 \end{pmatrix}$	$\begin{pmatrix} 4 & 5 \\ 3 & 7 \end{pmatrix}$	J48
$\begin{pmatrix} 9 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 3 & 6 \\ 2 & 8 \end{pmatrix}$	$\begin{pmatrix} 4 & 5 \\ 3 & 7 \end{pmatrix}$	J48graft
$\begin{pmatrix} 8 & 1 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 3 & 6 \\ 1 & 9 \end{pmatrix}$	$\begin{pmatrix} 4 & 5 \\ 3 & 7 \end{pmatrix}$	LMT

OneR has considerably decreased after LLSOM1–2 and UOM2 when the spline is approximated as the amplitude.

The performance of above methods (combination of feature extraction methods and classifiers) based on aforementioned statistical metrics is summarized in Table 5.9. It demonstrates LazyIB5, J48 and J48graft are performed well after LLSOM2-based preprocessing as their specificity and FNR are 1.000 and 0.000 respectively. These values show that no K-complex segments are misclassified as non-K-complexes. Further investigation requires to evaluate the performance of LazyIB5, J48 and J48graft after LLSOM2-based preprocessing.

Table 5.8: Confusion matrices (CMs) on the test set for the preprocessed dataset (after LLSOM1, LLSOM2, UOM1 and UOM2 when the spline ($m_1 = 4, n = 5$) is approximated as the amplitude).

CM				
LLSOM1	LLSOM2	UOM1	UOM2	Classifiers
$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	LibSVM
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}4 & 5\\1 & 9\end{pmatrix}$	$\begin{pmatrix}6 & 3\\6 & 4\end{pmatrix}$	$\begin{pmatrix}3 & 6\\1 & 9\end{pmatrix}$	Logistic
$\begin{pmatrix}7 & 2\\5 & 5\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	$\begin{pmatrix}3 & 6\\0 & 10\end{pmatrix}$	$\begin{pmatrix}0 & 9\\0 & 10\end{pmatrix}$	RBF Network
$\begin{pmatrix}7 & 2\\5 & 5\end{pmatrix}$	$\begin{pmatrix}8 & 1\\2 & 8\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}8 & 1\\8 & 2\end{pmatrix}$	SMO
$\begin{pmatrix}7 & 2\\5 & 5\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	$\begin{pmatrix}8 & 1\\8 & 2\end{pmatrix}$	$\begin{pmatrix}6 & 3\\1 & 9\end{pmatrix}$	LazyIB1
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}5 & 4\\0 & 10\end{pmatrix}$	$\begin{pmatrix}7 & 2\\8 & 2\end{pmatrix}$	$\begin{pmatrix}4 & 5\\1 & 9\end{pmatrix}$	LazyIB5
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	$\begin{pmatrix}6 & 3\\7 & 3\end{pmatrix}$	$\begin{pmatrix}4 & 5\\1 & 9\end{pmatrix}$	KStar
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	$\begin{pmatrix}6 & 3\\7 & 3\end{pmatrix}$	$\begin{pmatrix}6 & 3\\1 & 9\end{pmatrix}$	LWL
$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	OneR
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}5 & 4\\0 & 10\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	J48
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}5 & 4\\0 & 10\end{pmatrix}$	$\begin{pmatrix}9 & 0\\10 & 0\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	J48graft
$\begin{pmatrix}6 & 3\\2 & 8\end{pmatrix}$	$\begin{pmatrix}5 & 4\\1 & 9\end{pmatrix}$	$\begin{pmatrix}6 & 3\\6 & 4\end{pmatrix}$	$\begin{pmatrix}6 & 3\\1 & 9\end{pmatrix}$	LMT

Since there exist 39 and 31 segments containing non-K-complexes and K-complexes respectively there exists an unbalanced dataset. Therefore the area under ROC curve (ROC area) is required to assess the performance of a classifier regardless of class distribution. Its definition is given in Section 3.9 of Chapter 3. To evaluate the performance of the prominent classifiers their ROC areas are presented in Table 5.10. It demonstrates that which classifiers are performed well after LLSOM1, LLSOM2 and UOM2 in terms of the area under the ROC curve.

The results, as shown in Table 5.10, indicate that

Table 5.9: Performance of proposed methods based on corresponding statistical measures.

Method	Sensitivity	Specificity	FPR	FNR
LLSOM1+Logistic	0.444	0.9000	0.5556	0.100
LLSOM2+RBF Network	0.556	0.900	0.444	0.100
LLSOM2+SMO	0.889	0.800	0.111	0.200
UOM2+LazyIB1	0.667	0.900	0.333	0.100
LLSOM2+LazyIB5	0.556	1.000	0.444	0.000
LLSOM2+KStar	0.556	0.900	0.444	0.100
UOM2+LWL	0.667	0.9000	0.333	0.100
LLSOM2+J48	0.5565	1.000	0.444	0.000
LLSOM2+J48graft	0.556	1.000	0.444	0.000
UOM2+LMT	0.6667	0.900	0.333	0.100

- Logistic performs well after LLSOM1-based preprocessing with the ROC area of 0.731.
- RBF Network works better over the set of features obtained from LLSOM2 than the original dataset with the ROC area of 0.850
- LazyIB5 performs better than RBF Network, SMO, KStar, J48 and J48graft after LLSOM2-based preprocessing with the ROC area of 0.906.
- LWL executes better than LazyIB1 and LMT after UOM2-based processing with the value of 0.856 as the ROC area.
- KStar works better in combination with LLSOM2 than LLSOM1. In this case, the value of the ROC area is 0.839.

As can be seen from the Table 5.10, the nearest value to 1 is obtained from the combination of LLSOM2 with LazyIB5 (0.906). This value indicates that the proposed combination has high discriminating capability to classify an EEG signal between the sets of non-K-complexes and K-complexes. Further, it demonstrates that the LLSOM2 is a very promising feature extraction method in this particular application.

Example 5.4.1. ACC vs area under the ROC curve *These findings indicate that SMO and LazyIB5 are promising classifiers in combination with LLSOM2 in terms of ACC (84%) and the ROC area respectively. Confusion matrices obtained from SMO* *and LazyIB5 over the extracted features after LLSOM2-based preprocessing are as follows.*

1. *LLSOM2+SMO:* $\begin{pmatrix} 8 & 1 \\ 2 & 8 \end{pmatrix}$.
2. *LLSOM2+LazyIB5:* $\begin{pmatrix} 5 & 4 \\ 0 & 10 \end{pmatrix}$.

In the first confusion matrix, 1 *out of* 9 *non-K-complexes is categorized as the K-complex and* 2 *out of* 10 *K-complexes are categorized as non-K-complexes. Therefore*

- *ACC*$= 84\%$,
- *Sensitivity* $= 8/(8+1) = 0.889$,
- *Specificity* $= 8/(8+2) = 0.800$,
- *FPR* $= 1/9 = 0.111$,
- *FNR* $= 2/10 = 0.200$ *and*
- *The area under ROC curve* $= 0.844$.

In the second confusion matrix, 4 *out of* 9 *non-K-complexes are categorized as K-complexes and all* 10 *K-complexes are categorized correctly. Then,*

- *ACC*$= 79\%$,
- *Sensitivity* $= 5/(5+4) = 0.556$,
- *Specificity* $= 10/(0+10) = 1.000$,
- *FPR*$= 4/9 = 0.444$,
- *FNR*$= 0.000$ *and*
- *The area under ROC curve* $= 0.906$.

It is somewhat surprising that LazyIB5 performs better than SMO regardless of higher ACC obtained from SMO. Notwithstanding these ACCs, the area under ROC curve (ROC area) is often employed to assess the quality of the classifiers. A perfect classifier has a ROC area of one. As shown above, LazyIB5 has the higher values of specificity *and ROC area than SMO. Therefore LazyIB5 is a very promising classifier.*

Table 5.10: The **prominent results** of ROC area for (a) the original dataset, and (b) the preprocessed dataset when the spline ($m_1 = 4, n = 5$) is approximated as the amplitude.

ROC area for (a)	ROC area for (b)			Classifiers
	LLSOM1	LLSOM2	UOM2	
N/A	0.731			Logistic
0.815		0.850		RBF Network
0.500		0.844		SMO
0.511			0.822	LazyIB1
0.600		0.906		LazyIB5
0.500	0.733	0.839		KStar
0.806			0.856	LWL
0.500		0.778		J48
0.500		0.778		J48graft
0.472			0.783	LMT

The comparison of corresponding ROC areas presented in Table 5.10 indicates LazyIB5 has the highest value of the ROC area among others. Consequently, the combination of LLSOM2 (containing the spline) with LazyIB5 form a very promising method to detect K-complexes.

Tables 5.1 to 5.4 reveal that the mean frequencies are significantly higher for non-K-complexes. This observation is important for classification of an EEG signal in presence of K-complexes. Thus, we just keep ω as the only feature of an EEG signal instead of 24 and 45 features of LLSOM1, UOM1 and LLSOM2, UOM2 respectively. Note that the spline functions are more flexible and efficient than the polynomial ones to approximate the amplitude. Therefore ω is kept as the only extracted feature after optimization-based preprocessing approaches (LLSOM1, LLSOM2, UOM1 and UOM2) when the amplitude is approximated by splines.

The classification accuracy of K-complexes after optimization-based preprocessing approaches (OPAs) on the basis of frequency ω (one feature) is reported in Table 5.11. The classification accuracies are surprisingly high, especially in the case of LLSOM2, where we reach 79% for the first 12 classifiers. Consequently, a classifier can be developed based on the mean frequencies of K-complexes and non-K-complexes.

Appendix A describes a way to develop an additional classifier.

Table 5.11: Classification accuracy on the test set for the preprocessed dataset, one feature (ω).

Classifiers	LLSOM1	LLSOM2	UOM1	UOM2
LibSVM	74%	79%	53%	79%
Logistic	74%	79%	53%	74%
RBF Network	74%	79%	53%	74%
SMO	74%	79%	53%	74%
LazyIB1	74%	79%	53%	74%
LazyIB5	74%	79%	53%	79%
KStar	74%	79%	53%	74%
LWL	74%	79%	53%	74%
OneR	74%	79%	53%	74%
J48	74%	79%	47%	74%
J48graft	74%	79%	47%	74%
LMT	74%	79%	53%	79%

5.5 Summary

This research develops four convex optimization models to extract and generate the essential features of an EEG signal. In order to illustrate the efficiency and effectiveness of them an EEG K-complex dataset is taken as the recorded signal values. The spline and the polynomial functions are approximated as the amplitude. The results of this research confirm that the spline functions are more flexible and efficient then the polynomial ones to describe abrupt changes in the amplitude in order to detect the K-complexes. Note that one of the main properties of the K-complexes is a sudden change in the amplitude.

The numerical results demonstrate that the developed models enable robust and efficient detection of the K-complexes. Namely, they enable one to improve (in most cases) the classification accuracy after preprocessing while the computational time is not very high. The number of false negatives obtained from most of classifiers has considerably decreased after preprocessing.

Taken together, the main conclusions are as follows.

- OPAs enhance the classification accuracy of all 12 classifiers except LibSVM and RBF Network (for some models).
- LLSOM1–2 perform better with the spline than the polynomial.
- The best classification accuracy is achieved from the combination of LLSOM2 and SMO (45 features) with an accuracy of 84%.
- LazyIB5 is the highest quality classifier with the ROC area of 0.906 after LLSOM2-based preprocessing.
- Overall, LLSOM1 and LLSOM2 are fast and accurate. This may due to the fact that the corresponding optimization problems are much simpler than UOM1 and UOM2.
- UOM1 and UOM2 are so expensive. They will not be applied to large scale datasets.

Chapter 6

Large Scale Dataset of an EEG Signal for Seizure Detection

6.1 Introduction

This chapter is concerned with the capability of optimization-based preprocessing approaches (OPAs) introduced in Chapter 3 to extract and generate the key features of an EEG signal in the presence of seizures. Their performance is verified in numerical simulations of an automated detection of seizures problem. Epileptic seizures can be obstructed by detection of electrical changes in the brain that happen before the seizure takes place. Therefore the automatic detection of seizures is necessary due to the fact that the visual screening of EEG recordings is a time consuming task and requires experts to improve the diagnosis.

The efficiency of OPAs to extract the essential features of an EEG epileptic signal significantly affects the classification accuracy obtained from classifiers. To verify the improvements in the classification accuracy that can be achieved through employing OPAs, a different range of classifiers is applied. These classifiers are employed over the original dataset and a dataset after OPAs. Hence, this study is restricted to the development of feature extraction methods but not classifiers.

6.2 The Performance of OPAs

Four convex optimization models called LLSOM1, LLSOM2, UOM1 and UOM2 are developed in the feature extraction stage of the classification problem. The numerical results shown in Chapter 5 indicate that the last two models are computationally expensive and are not as efficient as the first two models to locate the abrupt changes in the amplitude. Consequently, they are not considered any further for large scale dataset.

LLSOM1 and LLSOM2 are based on a sequence of linear least squares problems. The analysis of them is provided in Chapter 4 that gives raise to the fact that the system matrices obtained from LLSOM1 and LLSOM2 are full-rank and rank-deficient respectively. Hence, the normal equations method is employed to solve LLSOM1 while an SVD method is used to solve LLSOM2. The polynomial (P) and the spline (S) functions described in Equations (3.2) and (3.4) of Chapter 3 respectively are used to model the signal amplitude. According to the technical specification of the EEG epileptic dataset provided in Section 3.10.2 of Chapter 3, $m_1 = 4$ (degree of a spline) and $n = 12$ (the number of subintervals) are assigned to the spline S. The knots are chosen as a sequence of equidistant knots. The frequency grid is specified as the numbers between $\omega_i = 0.53\,\text{Hz}$ and $\omega_f = 40\,\text{Hz}$ with the step size of $1\,\text{Hz}$ based on the given band-pass filter settings ($0.53 - 40\,\text{Hz}$ ($12\,\text{dB/oct}$, the unit dB/oct is deciBel per Octave where deciBel is a unit of logarithmic and an Octave is a doubling of frequency)) while the initial and final values of $\tau_i = 0$ and $\tau_f = \pi$ with the step size of $\pi/4$ are assigned to the phase shift grid.

In order to balance the number of parameters in a spline and a polynomial functions, the degree of P is increased to $m_2 = m_1 n$ where $m_1 = 4$ and $n = 12$. Then, $m_2 = 48$ is assigned to P. Herein, the results from LLSOM1 and LLSOM2 involving fixed knots splines S are comparable with ones involving polynomials P.

The values of objective function, ω and τ are considered as three additional essential features after LLSOM1 and LLSOM2. To simplify the notations, LLSOM1 and LLSOM2 involving P are denoted by LLSP1 and LLSP2 respectively while the ones involving S are denoted by LLSP3 and LLSP4 respectively.

The output dimensions of LLSP1, LLSP3 and LLSP2, LLSP4 are $m_2 + 1 = m_1 n + 1 = 49$ and $2m_2 + 2 = 2m_1 n + 2 = 98$ respectively. Therefore $N = 4097$ features of the original dataset have been reduced to $52 = m_2+4 = m_1 n+4$

and $101 = 2m_2 + 5 = 2m_1n + 5$ essential features after LLSP1, LLSP3 and LLSP2, LLSP4 respectively. Figure 5.1 of Chapter 5 illustrates the dimension of essential features (extracted features) after OPAs.

6.3 The Numerical Results and Discussion

In this section, the results obtained from LLSP1– 4 are presented in Tables 6.1 and 6.2. These tables give the computational time in seconds and the mean frequency for each set (A, B, C, D and E) of the EEG signal. The LLSP1– 4 enable one to

- extract key features of the waves that are crucial for detecting seizures,
- reduce the size of dataset,
- improve the performance of classifiers.

Table 6.1 indicates that LLSP2 and LLSP4 have higher CPU times than LLSP1 and LLSP3. The execution time of LLSP3 is less than LLSP1 therefore a preprocessing approach with a spline amplitude would be preferable in terms of computational time. LLSP3 had spent the least time to extract the essential features of a signal described in Experiment 1.

Table 6.2 displays the mean frequencies for each set of the EEG signal. One can see that the mean frequencies of seizures (set E) are significantly higher than those of non-seizure events (sets A, B, C and D) except for set B. It may be due to the fact that the frequency of seizures is higher than non-seizures. Therefore the frequency (ω) is not kept as the only feature of an EEG signal after LLSP1– 4 for the seizure detection unlike the K-complex detection. It is because of the fact that the pattern of EEG epileptic waveforms is not as clear as EEG K-complex waveforms.

The essential features of an EEG signal are extracted through LLSP1– 4 in the feature extraction stage. Therefore $N = 4097$ features of the original signal is reduced to 52 and 101 with 200 segments after employing LLSP1, LLSP3 and LLSP2, LLSP4 respectively.

Table 6.1: Computational time (in seconds) for preprocessing.

LLSP approaches	**LLSP1**	**LLSP2**	**LLSP3**	**LLSP4**
Experiment 1	2,209	5,652	1,562	4,219
Experiment 2	2,190	5,610	1,774	4,206
Experiment 3	2,223	5,702	1,723	4,458
Experiment 4	2,184	5,631	1,597	4,456

Table 6.2: Mean frequencies for each set of the EEG signal.

LLSP approaches	**LLSP1**	**LLSP2**	**LLSP3**	**LLSP4**
Set A	0.7500	1.9900	0.5400	1.9500
Set B	6.5800	6.9600	5.2500	9.6600
Set C	0.9800	1.4300	0.8000	1.1800
Set D	1.6800	1.7883	1.3300	1.9700
Set E	5.1100	5.0600	4.8500	4.9100

Four different binary classification problems made from sets A, B, C, D and E are used to validate LLSP1– 4 in order to detect seizures. They are described in Section 3.10.2 of Chapter 3 as Experiment 1 to Experiment 4.

To avoid inconsistency in EEG signals and enhance the performance of classifiers, the dataset related to each experiment is balanced. The proper balancing of datasets where there are the same number of segments in each class (seizure and non-seizure) is described in Section 3.10.2 of Chapter 3.

Several statistical measurements such as classification accuracy (ACC), precision, sensitivity, specificity, FPR and FNR are used to assess

- the robustness of LLSP1– 4 in extraction of essential features and
- the influence of LLSP1– 4 on the performance of classifiers.

Above metrics are described in Section 3.9 of Chapter 3. The area under the ROC curve is an alternative to the ACC when there exists an unbalanced dataset. Because the dataset of each experiment is balanced then, there is no need to consider the area under the curve for this particular application unlike the EEG K-complex dataset.

First, the classifiers are employed over the original dataset with $N = 4097$ features. The results are shown in the first column of Tables 6.3, 6.7, 6.11 and 6.15. The “Logistic”, “SMO” and “LMT” algorithms do not produce any results on original dataset. This is most probably due to the memory limitations of the used software implementation.

Second, all classifiers are applied over the obtained set of features after LLSP approaches. The classification accuracy results based on the four experiments are presented in the Tables 6.3, 6.7, 6.11 and 6.15.

The classification results obtained from four experiments after LLSP1– 4 are presented in the following sections.

6.3.1 Experiment 1

The first experiment contains five datasets (A, B, C, D and E) in such a way that sets A, B, C and D are treated as a non-seizure class while set E is treated as a seizure class. There exist 4097 features (recordings) and 200 segments such that the first 100 segments belongs to a non-seizure class whereas the last 100 segments to a seizure class. This experiment has 178 and 22 segments as training and test sets respectively. Therefore each of 12 classifiers is trained on the training set and tested on the test set and the ACC on the test set is reported.

Table 6.3 presents the ACC on the test set for the original dataset and the preprocessed dataset after LLSP1– 4. Because of the long computational time taken for LLSP4 (in Table 6.1) and the fact that the “RBF Network” classifier in Table 6.3 provides a better accuracy on the original dataset rather than the preprocessed dataset after LLSP4, the performance of LLSP4 is not satisfactory.

Although LLSP2 has a long computational time similar to LLSP4, “RBF Network” and “LibSVM” classifiers work well after LLSP2. Tables 6.3 shows that the accuracy of all classifiers is considerably improved and no classifiers failed on the preprocessed dataset. It is worth to note that some classifiers perform better with specific LLSP approaches.

- LibSVM and RBF Network work better after LLSP2;
- Logistic, SMO, LazyIB1, LazyIB5, KStar, LWL, J48, J48graft and LMT work well after LLSP3.

Confusion matrices based on the above specific LLSP approaches are provided in Table 6.4. The structure of a confusion matrix is expressed in Table 3.1 of Chapter 3. Their precision and sensitivity values are provided in Table 6.5 as well.

Table 6.3: Classification accuracy (ACC) of Experiment 1 on the test set for (a) the original dataset, 4097 features and (b) the preprocessed dataset (after LLSP1 to LLSP4), 52 features for LLSP1 and LLSP3, 101 features for LLSP2 and LLSP4.

ACC on (a)	ACC on (b)				Classifiers
	LLSP1	LLSP2	LLSP3	LLSP4	
45%	45%	50%	45%	45%	LibSVM
N/A [a]	95%	91%	100%	95%	Logistic
54%	68%	86%	68%	45%	RBF Network
N/A [a]	77%	91%	100%	95%	SMO
64%	77%	82%	100%	86%	LazyIB1
68%	73%	77%	91%	91%	LazyIB5
54%	82%	86%	100%	55%	KStar
68%	77%	86%	100%	86%	LWL
54%	55%	55%	55%	55%	OneR
45%	82%	91%	100%	95%	J48
45%	82%	91%	100%	95%	J48graft
N/A [a]	82%	91%	100%	91%	LMT

[a] No Answer.

In the confusion matrices presented in Table 6.4, entry $\{11\}$ corresponds to the number of non-seizures classified correctly, entry $\{12\}$ corresponds to the number of non-seizures classified as seizures (the number of false positives), entry $\{21\}$ corresponds to the number of seizures classified as non-seizures (the number of false negatives) and entry $\{22\}$ corresponds to the number of seizures classified correctly.

One of the main goals of feature extraction methods (LLSP1– 4) is to reduce the number of false negatives. Table 6.4 shows the rate of false negatives of all classifiers except LibSVM, J48 and J48graft is considerably decreased.

In summary, all classifiers except LibSVM and RBF Network are prominent in

Table 6.4: Confusion matrices of Experiment 1 from the **prominent combinations** of LLSP2 and LLSP3 with corresponding classifiers in terms of classification accuracy.

	Preprocessed dataset		
Original dataset	LLSP2	LLSP3	Classifiers
$\begin{pmatrix} 0 & 12 \\ 0 & 10 \end{pmatrix}$	$\begin{pmatrix} 1 & 11 \\ 0 & 10 \end{pmatrix}$		LibSVM
N/A [a]		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	Logistic
$\begin{pmatrix} 12 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 11 & 1 \\ 2 & 8 \end{pmatrix}$		RBF Network
N/A [a]		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	SMO
$\begin{pmatrix} 12 & 0 \\ 8 & 2 \end{pmatrix}$		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	LazyIB1
$\begin{pmatrix} 12 & 0 \\ 7 & 3 \end{pmatrix}$		$\begin{pmatrix} 12 & 0 \\ 2 & 8 \end{pmatrix}$	LazyIB5
$\begin{pmatrix} 12 & 0 \\ 10 & 0 \end{pmatrix}$		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	KStar
$\begin{pmatrix} 12 & 0 \\ 7 & 3 \end{pmatrix}$		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	LWL
$\begin{pmatrix} 0 & 12 \\ 0 & 10 \end{pmatrix}$		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	J48
$\begin{pmatrix} 0 & 12 \\ 0 & 10 \end{pmatrix}$		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	J48graft
N/A [a]		$\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$	LMT

[a] No Answer.

combination with LLSP3 with the zero value as the rate of false positives and negatives. Therefore LLSP3 is a very promising feature extraction method for Experiment 1. In addition, LLSP2 with RBF Network works well in spite of the long computational time for LLSP2 ($5,652$ seconds) presented in Table 6.1.

The precision and sensitivity values provided in Table 6.5 demonstrate which classifiers are performed well after LLSP2 and LLSP3. A perfect classifier has the precision and sensitivity values equal to 1. This value indicates that the combination of the proposed feature extraction method (LLSP2 and LLSP3) with the classifier has the high discriminating capability to classify an EEG signal between the sets of non-seizures and seizures. Hence, the best classifiers are Logistic, SMO, LazyIB1, KStar,

Table 6.5: Precision and Sensitivity values for the prominent classifiers in combination with LLSP2 and LLSP3 for Experiment 1.

Precision/Sensitivity	Precision/Sensitivity		Classifiers
Original dataset	LLSP2	LLSP3	
N/A [a]/ 0	1.00 / 0.08		LibSVM
N/A [a]		1.00 / 1.00	Logistic
0.55 / 1.00	0.85 / 0.92		RBF Network
N/A [a]		1.00 / 1.00	SMO
0.60 / 1.00		1.00 / 1.00	LazyIB1
0.63 / 1.00		0.86 / 1.00	LazyIB5
0.55 / 1.00		1.00 / 1.00	KStar
0.63 / 1.00		1.00 / 1.00	LWL
N/A [a]/ 0		1.00 / 1.00	J48
N/A [a]/ 0		1.00 / 1.00	J48graft
N/A [a]		1.00 / 1.00	LMT

[a] No Answer.

LWL, J48, J48graft and LMT after LLSP3 in terms of precision and sensitivity values. The precision and sensitivity values can be obtained from the corresponding confusion matrix.

Example 6.3.1. *The confusion matrices obtained from LazyIB1 on the original dataset and the dataset after LLSP3 are considered.*

1. *LLSP3+LazyIB1:* $\begin{pmatrix} 12 & 0 \\ 0 & 10 \end{pmatrix}$.

2. *Original dataset+LazyIB1:* $\begin{pmatrix} 12 & 0 \\ 8 & 2 \end{pmatrix}$.

There are 22 *segments in the test set where* 12 *segments belong to the non-seizure class and* 10 *segments to the seizure class. In the first confusion matrix,* 0 *out of* 12 *non-seizures is categorized as seizure and* 0 *out of* 10 *seizures are categorized as non-seizure. On the other hand, non-seizure and seizure segments are classified correctly. Therefore*

- *ACC*$= (12 + 10)/(12 + 0 + 0 + 10) = 1 \times 100 = 100\%$,
- *Sensitivity* $= 12/(12 + 0) = 1.00$,

- *Specificity* $= 10/(0+10) = 1.00$,
- *FPR* $= 0/(12+0) = 0.00$,
- *FNR* $= 0/(0+10) = 0.00$ *and*
- *Precision* $= 12/(12+0) = 1.00$.

In the second confusion matrix, all 12 *non-seizures are categorized correctly and* 8 *out of* 10 *seizures are categorized as non-seizures. Hence,*

- *ACC*$= (12+2)/(12+0+8+2) = 0.64 \times 100 = 64\%$,
- *Sensitivity* $= 12/(12+0) = 1.00$,
- *Specificity* $= 2/(8+2) = 0.20$,
- *FPR* $= 0/(12+0) = 0.00$,
- *FNR* $= 8/(8+2) = 0.80$ *and*
- *Precision* $= 12/(12+8) = 0.60$.

In summary, LazyIB1 works better on the preprocessed dataset after LLSP3 rather than the original dataset because the precision value of the first confusion matrix is 1 *whereas the second one is* 0.60*. The FNR that is also referred as type-II error is zero for the first confusion matrix while it is* 0.80 *for the second one.*

To evaluate the performance of corresponding classifiers with the accuracy of 100% (in Table 6.3) after LLSP3 their computational times are shown in Table 6.6. It illustrates which classifiers perform well after LLSP3 in terms of computational time. The performance of LMT with LLSP3 is not satisfactory since it has a long computational time (104 seconds). In conclusion, the combinations of Logistic and LazyIB1 with LLSP3 perform well for Experiment 1 with the classification accuracy of 100% and the computational time of 0.01 seconds.

Table 6.6: Computational time on the test set over the preprocessed dataset after LLSP3 for Experiment 1.

Classifiers	CPU time (in seconds)
Logistic	0.01
SMO	0.17
LazyIB1	0.01
KStar	0.05
LWL	0.06
J48	0.02
J48graft	0.18
LMT	104

6.3.2 Experiment 2

The second experiment contains four datasets (A, C, D and E). Sets A, C and D are treated as a non-seizure class while set E is treated as a seizure class. There are 4097 features (recordings) and 200 segments such that the first 100 segments belongs to a non-seizure class whereas the last 100 segments to a seizure class. This experiment has 180 and 20 segments as training and test sets respectively.

Table 6.7 demonstrates that the accuracy of all classifiers except LibSVM considerably improved after LLSP approaches (LLSP1− 4) rather than the original dataset. Although the LibSVM classifier provides a better accuracy on the original dataset, no classification method failed on the preprocessed dataset after LLSP approaches. Most of the classifiers in Table 6.7 achieved the accuracy of 100% after LLSP1.

Since the maximum accuracy obtained after LLSP4 is 95% and its computational time reported in Table 6.1 is $4,206$ seconds then, the performance of LLSP4 in Experiment 2 is not satisfactory. Moreover, the performance of LLSP2 is not efficient regardless of 100% accuracy obtained from Logistic because of the long computational time ($5,6104$ seconds) presented in Table 6.1.

Some classifiers in Table 6.7 carry out better with specific LLSP approaches.

- Logistic, J48, J48graft and LMT work well after LLSP1;
- SMO, LazyIB1, KStar and LWL perform well after LLSP1 and LLSP3;

Table 6.7: Classification accuracy (ACC) of Experiment 2 on the test set for (a) the original dataset, 4097 features and (b) the preprocessed dataset (after LLSP1 to LLSP4), 52 features for LLSP1 and LLSP3, 101 features for LLSP2 and LLSP4.

ACC on (a)	ACC on (b)				Classifiers
	LLSP1	LLSP2	LLSP3	LLSP4	
85%	50%	55%	60%	50%	LibSVM
N/A [a]	100%	100%	95%	95%	Logistic
50%	85%	75%	90%	50%	RBF Network
N/A [a]	100%	85%	100%	95%	SMO
65%	100%	80%	100%	80%	LazyIB1
65%	95%	70%	100%	85%	LazyIB5
50%	100%	80%	100%	50%	KStar
65%	100%	90%	100%	95%	LWL
50%	50%	50%	50%	50%	OneR
50%	100%	95%	95%	95%	J48
50%	100%	95%	95%	95%	J48graft
N/A [a]	100%	85%	95%	91%	LMT

[a] No Answer.

- RBF Network and LazyIB5 work well after LLSP3.

Their confusion matrices and precision/sensitivity values are shown in Tables 6.8 and 6.9 respectively. Table 6.8 indicates that the rate of false negatives of all classifiers is considerably decreased. Therefore all classifiers except RBF Network and LazyIB5 are prominent in combination with LLSP1. Further, all classifiers except Logistic, J48, J48graft and LMT work well after LLSP3.

The precision and sensitivity values presented in Table 6.9 illustrate that the best classifiers after LLSP1 and LLSP3 are SMO, LazyIB1, KStar and LWL with the value of 1 for both precision and sensitivity. In addition, Logistic, J48, J48graft and LMT are the best classifiers after LLSP1 while LazyIB5 is the best classifier after LLSP3 with precision and sensitivity values of 1.

To investigate the performance of the corresponding classifiers after LLSP1 and LLSP3 their computational times are reported in Table 6.10. As discussed above, most of classifiers reached high accuracy of 100% after LLSP1 of Experiment 2. So, a preprocessing approach with a polynomial amplitude (LLSP1) is preferable.

Table 6.8: Confusion matrices of Experiment 2 from the prominent combinations of LLSP1 and LLSP3 with corresponding classifiers in terms of classification accuracy.

	Preprocessed dataset		
Original dataset	LLSP1	LLSP3	Classifiers
N/A [a]	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$		Logistic
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$		$\begin{pmatrix}9 & 1\\ 1 & 9\end{pmatrix}$	RBF Network
N/A [a]	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	SMO
$\begin{pmatrix}10 & 0\\ 7 & 3\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LazyIB1
$\begin{pmatrix}10 & 0\\ 7 & 3\end{pmatrix}$		$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LazyIB5
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	KStar
$\begin{pmatrix}10 & 0\\ 7 & 3\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LWL
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$		J48
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$		J48graft
N/A [a]	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$		LMT

[a] No Answer.

Taken together, the results from Table 6.10 suggest that for Experiment 2 the combinations of LazyIB1 and J48 with LLSP1 perform well with the classification accuracy of 100% and the zero value of computational time. The combination of LazyIB5 with LLSP3 results in the classification accuracy of 100% and the computational time of 0.01 seconds.

Table 6.9: Precision and Sensitivity values for the prominent classifiers in combination with LLSP1 and LLSP3 for Experiment 2.

Precision/Sensitivity	Precision/Sensitivity		Classifiers
Original dataset	LLSP1	LLSP3	
N/A [a]	1.00 / 1.00		Logistic
0.50 / 1.00		0.90 / 0.90	RBF Network
N/A [a]	1.00 / 1.00	1.00 / 1.00	SMO
0.60 / 1.00	1.00 / 1.00	1.00 / 1.00	LazyIB1
0.60 / 1.00		1.00 / 1.00	LazyIB5
0.50 / 1.00	1.00 / 1.00	1.00 / 1.00	KStar
0.60 / 1.00	1.00 / 1.00	1.00 / 1.00	LWL
0.50 / 1.00	1.00 / 1.00		J48
0.50 / 1.00	1.00 / 1.00		J48graft
N/A [a]	1.00 / 1.00		LMT

[a] No Answer.

Table 6.10: Computational time on the test set over the preprocessed dataset after LLSP1 and LLSP3 for Experiment 2.

Classifiers with LLSP1	**CPU time (in seconds)**
Logistic	0.03
SMO	0.22
LazyIB1	0
KStar	0.05
LWL	0.05
J48	0
J48graft	0.12
LMT	89
Classifiers with LLSP3	**CPU time (in seconds)**
SMO	0.15
LazyIB1	0.01
LazyIB5	0.01
KStar	0.13
LWL	0.16

6.3.3 Experiment 3

This experiment includes two datasets called B and E. Set B belongs to a non-seizure class and set E belongs to a seizure class. There exist 4097 features (recordings) and 200 segments in such a way that the first 100 segments belongs to a non-seizure class while the last 100 segments to a seizure class. Similar to Experiment 2, Experiment 3 has 180 and 20 segments as training and test sets respectively.

All classifiers in Table 6.11 except LibSVM have achieved the better classification accuracy on the preprocessed dataset than original one. Most of the classifiers obtained the accuracy of 100% after LLSP1. The maximum classification accuracy obtained after LLSP2 is 95% and it has a long computational time of $5,702$ seconds (Table 6.1). Although LLSP2 is not a suitable preprocessing approach for Experiment 3, RBF Network performs well after it.

LLSP4 is not a better suited method for preprocessing since it has a long computational time of $4,458$ seconds (Table 6.1) in spite of the obtained classification accuracy of 100% for Logistic and LMT. LibSVM gives the accuracy of 55% on the original dataset and after LLSP3. So, LibSVM works better after LLSP3. There are classifiers that perform well with specific LLSP approaches as follows:

- Logistic, SMO and LMT perform well after LLSP1 and LLSP3;
- RBF Network and LWL work better after LLSP2;
- LazyIB1, LazyIB5, KStar, J48 and J48graft work well after LLSP1.

Confusion matrices and precision/sensitivity values of all above specific LLSPs are illustrated in Tables 6.12 and 6.13 respectively. As shown in Table 6.12, the rate of false negatives of all classifiers is considerably decreased.

It is apparent from Table 6.13 that the best classifiers with the precision and sensitivity values of 1 are as follows:

- Logistic, SMO and LMT are the best classifiers after LLSP1 and LLSP3;
- LazyIB1, KStar, J48 and J48graft are the best classifiers after LLSP1.

Table 6.11: Classification accuracy (ACC) of Experiment 3 on the test set for (a) the original dataset, 4097 features and (b) the preprocessed dataset (after LLSP1 to LLSP4), 52 features for LLSP1 and LLSP3, 101 features for LLSP2 and LLSP4.

ACC on (a)	ACC on (b)				Classifiers
	LLSP1	LLSP2	LLSP3	LLSP4	
55%	50%	50%	55%	50%	LibSVM
N/A [a]	100%	95%	100%	100%	Logistic
50%	50%	90%	65%	70%	RBF Network
N/A [a]	100%	95%	100%	95%	SMO
65%	100%	95%	90%	95%	LazyIB1
60%	95%	85%	90%	95%	LazyIB5
50%	100%	95%	90%	50%	KStar
65%	90%	95%	80%	95%	LWL
50%	50%	50%	50%	50%	OneR
50%	100%	95%	95%	95%	J48
50%	100%	95%	95%	95%	J48graft
N/A [a]	100%	95%	100%	100%	LMT

[a] No Answer.

To evaluate the performance of LLSP1 and LLSP3 with the corresponding classifiers that obtained the classification accuracy of 100%, the values of computational time are set out in Table 6.14. It is apparent from this table that the combinations of Logistic and LazyIB1 with LLSP1, and Logistic with LLSP3 perform well with the classification accuracy of 100%. Interestingly, LLSP1 is a better suited approach for Experiment 3 since most of classifiers achieved the maximum accuracy of 100% in combination with it.

Table 6.12: Confusion matrices of Experiment 3 from the prominent combinations of LLSP1, LLSP2 and LLSP3 with corresponding classifiers in terms of classification accuracy.

	Preprocessed dataset			
Original dataset	LLSP1	LLSP2	LLSP3	Classifiers
N/A [a]	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$		$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$	Logistic
$\begin{pmatrix} 10 & 0 \\ 10 & 0 \end{pmatrix}$		$\begin{pmatrix} 10 & 0 \\ 2 & 8 \end{pmatrix}$		RBF Network
N/A [a]	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$		$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$	SMO
$\begin{pmatrix} 10 & 0 \\ 7 & 3 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$			LazyIB1
$\begin{pmatrix} 10 & 0 \\ 8 & 2 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ 1 & 9 \end{pmatrix}$			LazyIB5
$\begin{pmatrix} 10 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$			KStar
$\begin{pmatrix} 10 & 0 \\ 7 & 3 \end{pmatrix}$		$\begin{pmatrix} 10 & 0 \\ 1 & 9 \end{pmatrix}$		LWL
$\begin{pmatrix} 10 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$			J48
$\begin{pmatrix} 10 & 0 \\ 10 & 0 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$			J48graft
N/A [a]	$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$		$\begin{pmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix}$	LMT

[a] No Answer.

Table 6.13: Precision and Sensitivity values for the prominent classifiers in combination with LLSP1, LLSP2 and LLSP3 for Experiment 3.

Precision/Sensitivity	Precision/Sensitivity			Classifiers
Original dataset	LLSP1	LLSP2	LLSP3	
N/A [a]	1.00 / 1.00		1.00 / 1.00	Logistic
0.50 / 1.00		0.83 / 1.00		RBF Network
N/A [a]	1.00 / 1.00		1.00 / 1.00	SMO
0.59 / 1.00	1.00 / 1.00			LazyIB1
0.55 / 1.00	0.91 / 1.00			LazyIB5
0.50 / 1.00	1.00 / 1.00			KStar
0.59 / 1.00		0.91 / 1.00		LWL
0.50 / 1.00	1.00 / 1.00			J48
0.50 / 1.00	1.00 / 1.00			J48graft
N/A [a]	1.00 / 1.00		1.00 / 1.00	LMT

[a] No Answer.

Table 6.14: Computational time on the test set after LLSP1 and LLSP3 for Experiment 3.

Classifiers with LLSP1	**CPU time (in seconds)**
Logistic	0.01
SMO	0.12
LazyIB1	0.01
KStar	0.05
J48	0.02
J48graft	0.45
LMT	124
Classifiers with LLSP3	**CPU time (in seconds)**
Logistic	0.01
SMO	0.15
LMT	138

6.3.4 Experiment 4

Similar to Experiment 3, the last experiment contains two datasets called sets A and E. Set A is treated as a non-seizure class whereas set E is treated as a seizure class. There exist 200 segment where the first 100 segments are considered as a non-seizure class and the second 100 segments as a seizure class. There are 180 and 20 segments as training and test sets respectively same as Experiment 3.

Table 6.15 presents the classification accuracy of Experiment 4 on the original and preprocessed datasets (after LLSP approaches). As it can be seen from the table below LibSVM provides a better accuracy on the original dataset rather than the preprocessed dataset. The performances of LLSP2 and LLSP4 are not satisfactory due to their long computational times for Experiment 4.

Table 6.15: Classification accuracy (ACC) of Experiment 4 on the test set for (a) the original dataset, 4097 features and (b) the preprocessed dataset (after LLSP1 to LLSP4), 52 features for LLSP1 and LLSP3, 101 features for LLSP2 and LLSP4.

ACC on (a)	ACC on (b)				Classifiers
	LLSP1	LLSP2	LLSP3	LLSP4	
75%	50%	55%	65%	55%	LibSVM
N/A [a]	100%	90%	100%	90%	Logistic
50%	50%	85%	85%	50%	RBF Network
N/A [a]	100%	90%	100%	95%	SMO
65%	100%	85%	100%	85%	LazyIB1
65%	95%	85%	100%	80%	LazyIB5
50%	100%	85%	100%	50%	KStar
65%	100%	80%	100%	95%	LWL
55%	50%	50%	50%	50%	OneR
50%	100%	90%	95%	90%	J48
50%	100%	90%	95%	90%	J48graft
N/A [a]	100%	90%	100%	95%	LMT

[a] No Answer.

The more surprising is with the simpler preprocessing approaches called LLSP1 and LLSP3. They obtained the highest classification accuracy of 100% in combination with the most of classifiers. They are faster than LLSP2 and LLSP4. In summary, some classifiers work well with specific preprocessing approaches.

- Logistic, SMO, LazyIB1, KStar, LWL and LMT perform well after LLSP1 and LLSP3;
- RBF Network and LazyIB5 work well after LLSP3;
- J48 and J48graft perform well after LLSP1.

Their confusion matrices and precision/sensitivity values are displayed in Tables 6.16 and 6.17 respectively. Table 6.16 demonstrates that the rate of false negatives of all classifiers is considerably decreased.

Table 6.16: Confusion matrices of Experiment 4 from the prominent combinations of LLSP1 and LLSP3 with corresponding classifiers in terms of classification accuracy.

	Preprocessed dataset		
Original dataset	LLSP1	LLSP3	Classifiers
N/A [a]	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	Logistic
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$		$\begin{pmatrix}7 & 3\\ 0 & 10\end{pmatrix}$	RBF Network
N/A [a]	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	SMO
$\begin{pmatrix}10 & 0\\ 7 & 3\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LazyIB1
$\begin{pmatrix}10 & 0\\ 7 & 3\end{pmatrix}$		$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LazyIB5
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	KStar
$\begin{pmatrix}10 & 0\\ 7 & 3\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LWL
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$		J48
$\begin{pmatrix}10 & 0\\ 10 & 0\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$		J48graft
N/A [a]	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	$\begin{pmatrix}10 & 0\\ 0 & 10\end{pmatrix}$	LMT

[a] No Answer.

Table 6.17 indicates that the best classifiers in combination with LLSP1 and LLSP3 with precision and sensitivity values of 1 are as follows.

- Logistic, SMO, LazyIB1, KStar, LWL and LMT after LLSP1 and LLSP3;

Table 6.17: Precision and Sensitivity values for the prominent classifiers in combination with LLSP1 and LLSP3 for Experiment 4.

Precision/Sensitivity	Precision/Sensitivity		Classifiers
Original dataset	LLSP1	LLSP3	
N/A [a]	1.00 / 1.00	1.00 / 1.00	Logistic
0.50 / 1		0.89 / 0.80	RBF Network
N/A [a]	1.00 / 1.00	1.00 / 1.00	SMO
0.59 / 1.00	1.00 / 1.00	1.00 / 1.00	LazyIB1
0.59 / 1.00		1.00/ 1.00	LazyIB5
0.50 / 1.00	1.00 / 1.00	1.00 / 1.00	KStar
0.59 / 1.00	1.00 / 1.00	1.00 / 1.00	LWL
0.50 / 1.00	1.00 / 1.00		J48
0.50 / 1.00	1.00 / 1.00		J48graft
N/A [a]	1.00 / 1.00	1.00 / 1.00	LMT

[a] No Answer.

- J48 and J48 graft after LLSP1;
- LazyIB5 after LLSP3.

To assess the performance of LLSP1 and LLSP3 in combination with the corresponding classifiers with the accuracy of 100%, Table 6.18 is presented. Logistic and LazyIB5 perform well after LLSP1 and LLSP3 respectively while LazyIB1 performs well after both LLSP1 and LLSP3.

Table 6.18: Computational time on the test set after LLSP1 and LLSP3 for Experiment 4.

Classifiers with LLSP1	**CPU time (in seconds)**
Logistic	0.01
SMO	0.14
LazyIB1	0.01
KStar	0.07
LWL	0.06
J48	0.02
J48graft	0.34
LMT	90
Classifiers with LLSP3	**CPU time (in seconds)**
Logistic	0.03
SMO	0.13
LazyIB1	0.01
LazyIB5	0.01
KStar	0.05
LWL	0.15
LMT	96

6.4 Comparative Performance of LLSP Approaches with Classifiers

The performance of above methods (combinations of feature extraction models and classification algorithms) based on the aforementioned statistical metrics is summarized in Table 6.19. Therefore no seizure segments are misclassified as non-seizure and vice versa. A comparison of the related algorithms' performance in detection of epileptic seizures is presented in Table 6.20 in terms of the classification accuracy (ACC). Further, it should be noted that the total accuracy of the proposed methods in this work is improved in the case of all experiments.

Table 6.19: Performance of proposed methods based on corresponding statistical measures.

Method	Experiment	Sensitivity	Specificity	FPR	FNR
LLSP3+Logistic, LazyIB1	1	1	1	0	0
LLSP1+LazyIB1, J48	2	1	1	0	0
LLSP3+LazyIB5	2	1	1	0	0
LLSP1+Logistic, LazyIB1	3	1	1	0	0
LLSP3+Logistic	3	1	1	0	0
LLSP1+Logistic, LazyIB1	4	1	1	0	0
LLSP3+LazyIB1, LazyIB5	4	1	1	0	0

Table 6.20: Comparative performance based on the classification accuracy obtained by various methods.

Method	Experiment	ACC (%)
Time frequency analysis-artificial neuralnetwork [91]	1	97.73
Multi-wavelet transform and approximate entropy feature-MLPNN [38]	1	98.27
Wavelet-based sparse functional linear model [98]	1	100
DE-RBFNs ensemble [25]	1	97.60
LLSP3-Logistic, LazyIB1 (this work)	1	100
DE-RBFNs ensemble [25]	2	99.25
Discrete wavelet transform-approximate entropy (ApEn) [66]	2	96.65
Discrete wavelet transform-line length feature-MLPNN [37]	2	97.75
LLSP1-LazyIB1, J48 (this work)	2	100
Time frequency analysis [92]	3	94.50
EEG complexity and spectral analysis [54]	3	98.33
Sample entropy and extreme learning machine [79]	3	95.67
Spectral method and statistical analysis [71]	3	97.50
LLSP1-Logistic, LazyIB1 (this work)	3	100
LLSP3-Logistic (this work)	3	100
Nonlinear pre-processing filter-Diagnostic neural network [60]	4	97.2
Time frequency domain features-Recurrent neural network [80]	4	99.6
Entropy measures-Adaptive neuro-fuzzy inference system [47]	4	92.22
Chaotic measures-Surrogate data analysis [46]	4	90
Fast Fourier transform-Decision tree [69]	4	98.72
Discrete wavelet transform-Mixture of expert model [82]	4	95
Time frequency analysis-Artificial neural network [91]	4	100
Discrete wavelet transform-relative wavelet energy-MLPNN [36]	4	95.2
Discrete wavelet transform-line length feature-MLPNN [37]	4	99.6
ANN methods [17]	4	98.3
LLSP1-Logistic, LazyIB1 (this work)	4	100
LLSP3-LazyIB1, LazyIB5 (this work)	4	100

6.5 Summary

An epileptic EEG signal has been approximated by a sine wave. Its amplitude is approximated by a polynomial of increased degree and a spline. The parameters of the amplitude are optimized by solving a sequence of LLSPs through a normal equations method if the system matrix is full-rank. An SVD method is employed to solve a sequence of LLSPs if its system matrix is rank-deficient.

The preprocessing approaches (LLSP1– 4) are used to extract the key features of an epileptic EEG signal. Four different experiments are carried out to evaluate the performance of the preprocessing models in the classification of an EEG signal. A promising performance is reported based on the evaluation criteria described in Section 3.9 of Chapter 3.

The findings of this study are summarized below. Following combinations achieved the classification accuracy of 100%.

1. Logistic and LazyIB1 perform well with LLSP3 for Experiment 1;
2. LazyIB1 and J48 work well with LLSP1, and LazyIB5 performs well with LLSP3 for Experiment 2;
3. Logistic performs well with LLSP1 and LLSP3, and LazyIB1 works well with LLSP1 for Experiment 3;
4. Logistic and LazyIB5 work well with LLSP1 and LLSP3 respectively, and LazyIB1 performs well with both LLSP1 and LLSP3 for Experiment 4.

Generally, LLSP1 and LLSP3 are fast and accurate feature extraction methods since they are much simpler than LLSP2 and LLSP4. The best classifiers for this work are Logistic, LazyIB1, LazyIB5 and J48. The numerical results show that most of classifiers achieved the classification accuracy of 100% after LLSP1 except for Experiment 1 where LLSP3 works well. Therefore LLSP1 carries out better in terms of classification accuracy whereas LLSP3 performs well in terms of computational time for this particular application.

Chapter 7

Summary and Conclusions

7.1 Summary

This chapter presents a summary of the work which was done throughout this study. EEG measures and records human brain activities. It has an important role to help in the diagnosis of brain diseases such as epileptic seizures. The necessity of classification of an EEG signal is clear in biomedical research since recording of a brain activity results in obtaining a very large and a complex set of data. Identification of different types of EEG waveforms is a complicated problem. It needs the analysis of large sets of EEG signals and requires a long computational time.

In classification of EEG signals, representative features of EEG recordings play a vital role. One interesting finding is that if the extracted features from a signal are not accurate enough to describe the original signal, the classification algorithms do not recognize those features appropriately. Hence, the quality of the performance of classification algorithms depends on the efficiency of the extracted features. Consequently, the development of sophisticated feature extraction methods can significantly enhance the performance of classification algorithms. This research intends to develop convex optimization-based models such that an EEG signal is approximated by a sine wave and its amplitude is approximated by a polynomial of increased degree and a spline. Developed models are used as feature extractors in order to detect transient events called K-complexes and seizures automatically.

This thesis discusses the significance of convex optimization-based methods to extract and generate the essential features of EEG signals in order to reduce the dimensionality of the recording data and improve the performance of classification algorithms. Four feature extraction methods are developed based on convex optimization problems. The first two methods (LLSOM1 and LLSOM2) are linear least squares problems (LLSPs) while the last two ones are uniform approximation problems (UOM1 and UOM2).

Most LLSPs can be solved using the system of normal equations if the corresponding matrix is non-singular otherwise one needs to apply a more robust (and time-consuming) approach (for example, QR decomposition and SVD). To identify when the corresponding matrix is non-singular, the singularity verification rules are developed for the first two methods (LLSOM1 and LLSOM2). Therefore one can choose a more suitable method for solving the corresponding LLSPs and enhance the efficiency of the approximation algorithms. This issue is especially important when one needs to solve the corresponding problems repeatedly. This is the case for our approximation algorithm where we run the experiments by assigning different values to ω and τ rather than optimizing them (due to the corresponding optimization problem complexity).

Consequently, the parameters of the approximated amplitudes are optimized by solving a sequence of LLSOM1 through a normal equations method since the system matrix is full-rank. An SVD method is employed to solve a sequence of LLSOM2 since its system matrix is rank-deficient.

The linear programming reformulation of last two methods (UOM1 and UOM2) that are non-smooth convex optimization problems are provided to verify that their solutions are optimal. They are solved using the CVX software and the LINPROG solver in MATLAB in order to optimize the parameters of the amplitude and extract the essential features of an EEG signal.

After extracting the essential features of an EEG signal through developed models, a range of different classifiers from WEKA is applied over the set of extracted features. These classifiers are used to evaluate the performance of developed models in classification of an EEG signal.

The K-complex and epileptic seizure datasets are applied to validate the proposed models. Further, they are compared based on the different statistical measures like ACC, TPR, TNR, FPR, FNR, the area under the ROC curve and corresponding computational time. Then, efficient and robust methods, best classifiers, and the best combinations of proposed methods with classifiers are reported for each dataset separately.

7.2 Conclusions

The numerical results demonstrate that the developed models, namely LLSOM1, LLSOM2, UOM1 and UOM2, are robust and efficient for automatic detection of K-complexes. They enable us to improve (in most cases) the classification accuracies after preprocessing, while the computational time is not very high. The main conclusions for the K-complex detection are as follows.

1. Optimization-based preprocessing approaches (OPAs) improve the classification accuracy of all 12 classifiers, except LibSVM and RBF (for some models).
2. We observe that most classifiers produce similar classification accuracies when only one feature (frequency ω) is used in the classification stage after OPAs (see Table 5.11). Therefore the frequency (ω) is an essential feature for all classification methods.
3. LLSOM1 and LLSOM2 perform better with splines than polynomials.
4. The combination of LLSOM2 and SMO (45 features) with an accuracy of 84% gives the best classification accuracy result when a spline is approximated as an amplitude.
5. The highest quality classifier is LazyIB5 after LLSOM2 (involving a spline) in terms of the area under the ROC curve with the value of 0.906.
6. LLSOM1 and LLSOM2 are fast and accurate since they are much simpler than UOM1 and UOM2.
7. UOM1 and UOM2 are computationally expensive therefore they are not efficient to apply over large scale datasets such as epileptic seizure datasets.

The singularity of the system matrices obtained from LLSOM1 and LLSOM2 are analyzed. The main difference between these two models is that in LLSOM2, the signal is shifted vertically (signal biasing) by a polynomial and a spline. The corresponding optimization problems can be formulated as linear least squares problems (LLSPs). If the system matrix is non-singular, then, the corresponding problem can be solved inexpensively and efficiently while for singular cases, slower (but more robust) methods have to be used. To choose a better suited method for solving the corresponding LLSPs we have developed singularity verification rules. In this thesis, we develop necessary and sufficient conditions for non-singularity of LLSOM1, and sufficient conditions for non-singularity and singularity of LLSOM2. Consequently, the system matrices obtained from LLSOM1 and LLSOM2 are full-rank and rank-deficient respectively. These conditions can be verified much faster than the direct singularity verification of the system matrices. Therefore the algorithm efficiency can be improved by choosing a suitable method for solving the corresponding LLSPs.

Since UOM1 and UOM2 are not as efficient as LLSOM1 and LLSOM2 for large scale datasets then, they are not considered for the EEG epileptic datasets. The preprocessing approaches (developed models), namely LLSOM1 and LLSOM2, are used to extract the essential features of an epileptic EEG signal. For simplicity, LLSOM1 with a polynomial and a spline as the amplitudes is referred to LLSP1 and LLSP3 respectively. In addition, LLSOM2 with a polynomial and a spline is referred to LLSP2 and LLSP4 respectively. Four different experiments are carried out to evaluate the performance of the preprocessing models in the classification of an EEG signal in presence of seizures.

In the study of signal approximation, the splines with fixed knots are preferable to the higher degree polynomials in order to approximate an amplitude, due to the fact that

- higher degree polynomials are unstable functions,
- splines are suitable candidates to describe abrupt changes in the amplitude and
- OPAs involving splines had spent the least time to extract the essential features of an EEG signal.

Subsequently, LLSOM1 and LLSOM2 (LLSP1 and LLSP3) are promising models to generate and extract the essential features of an EEG signal. The main conclusions for classification of an epileptic EEG signal are as follows.

- LLSP1 and LLSP3 are fast and accurate feature extraction methods because they are much simpler than LLSP2 and LLSP4.
- The best classifiers are Logistic, LazyIB1, LazyIB5 and J48.
- Most of classifiers achieved the ACC of 100% after LLSP1 except for Experiment 1 where LLSP3 performs well.
- In Experiment 1, LLSP3 is a very promising feature extraction method.
- In Experiments 2 to 4, LLSP1 performs better in terms of classification accuracy while LLSP3 carries out better in terms of computational time.

7.3 Suggestions for Future Work

This research work can be extended along the following research directions.

- Study the shape of the residuals to be able to find a suitable function for the vertical shift approximation.
- Further development of necessary (if possible) and sufficient conditions for non-singularity of LLSOM2.
- The development of more flexible models where vertical shift (signal biasing) splines do not have the same degrees and knot locations as they have in the main spline.
- Develop analytical optimality conditions to UOM2. The optimality conditions for UOM1 is developed by Sukhorukova and Ugon in 2016 [85].
- Develop a classifier based on the mean frequencies for targeted and non-targeted classes. More detail is provided in Appendix A.
- The extension of the results to the case when other types of functions (not necessary polynomial splines) are used to construct the corresponding approximations.

Bibliography

[1] H. Adeli, Z. Zhou, and N. Dadmehr. Analysis of EEG records in an epileptic patient using wavelet transform. *Journal of Neuroscience Methods*, 123(1):69 – 87, 2003. ISSN 0165-0270. doi: http://dx.doi.org/10.1016/S0165-0270(02)00340-0.

[2] R. Agarwal and J. Gotman. Digital tools in polysomnography. *Journal of Clinical Neurophysiology*, 19(2):136–143, March 2002.

[3] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: dependence on recording region and brain state. *Physical Review E.*, 64:061907, 2001. doi: 10.1103/PhysRevE.64.061907.

[4] V. Bajaj and R. B. Pachori. Classification of seizure and nonseizure eeg signals using empirical mode decomposition. *Information Technology in Biomedicine, IEEE Transactions on*, 16(6):1135–1142, Nov 2012. ISSN 1089-7771. doi: 10.1109/TITB.2011.2181403.

[5] W. W. Rouse Ball. *A Short Account of the History of Mathematics*. Dover publications, Inc., Mineola, NY, USA, 4nd edition, 1908.

[6] I. N. Bankman, V. G. Sigillito, R. A. Wise, and P. L. Smith. Feature-based detection of the k-complex wave in the human electroencephalogram using neural networks. *Biomedical Engineering, IEEE Transactions on*, 39(12): 1305–1310, 1992.

[7] J. L. Barlow. Numerical aspects of solving linear least squares problems. Technical report, Computer Science Department, The Pennsylvania State University, University Park, PA, USA, January 1999.

[8] A. Bhardwaj, A. Tiwari, R. Krishna, and V. Varma. A novel genetic programming approach for epileptic seizure detection. *Computer Methods and Programs in Biomedicine*, pages –, 2015. ISSN 0169-2607. doi: http://dx.doi.org/10.1016/j.cmpb.2015.10.001. URL `http://www.sciencedirect.com/science/article/pii/S016926071500262X`.

[9] A. Bjõrck. *Numerical Methods for Least Squares Problems*. Handbook of Numerical Analysis. Society for Industrial and Applied Mathematics, 1996.

[10] A. Bjorck. The calculation of linear least squares problems. *Acta Numerica*, 13:1–53, 4 2004.

[11] P. Borwein, I. Daubechies, V. Totik, and G. Nürnberger. Bivariate segment approximation and free knot splines: Research problems 96-4. *Constructive Approximation*, 12(4):555–558, 1996.

[12] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2010.

[13] G. Bremer, J. R. Smith, and I. Karacan. Automatic detection of the k-complex in sleep electroencephalograms. *Biomedical Engineering, IEEE Transactions on*, BME-17(4):314–323, 1970. ISSN 0018-9294. doi: 10.1109/TBME.1970.4502759.

[14] R. Brunelli. *Template Matching Techniques in Computer Vision: Theory and Practice*. Wiley Publishing, 2009. ISBN 0470517069, 9780470517062.

[15] P. R. Carney, R. B. Berry, and J. D. Geyer. *Clinical Sleep Disorders*. LWW medical book collection. Lippincott Williams & Wilkins, 2005. ISBN 9780781746373. `http://www.lww.com/Product/9780781786928`.

[16] G. D. Cetin, O. Cetin, and M. R. Bozkurt. Article: The detection of normal and epileptic eeg signals using ann methods with matlab-based gui. *International Journal of Computer Applications*, 114(12):45–50, March 2015.

[17] G. D. Cetin, O. Cetin, and M. R. Bozkurt. The detection of normal and epileptic EEG signals using ANN methods with matlab-based gui. *International Journal of Computer Applications*, 114(12):45–50, 2015. doi: 10.5120/20034-2145.

[18] N. G. Chebotarev. *On a certain minimax criterion*. Dokl. Akad. Nauk SSSR 39, 373–376 (see also Collected Works Vol. 2, Moscow, Izdatel'stvo Akad, Nauk SSSR, 1949), 1943.

[19] A. Cohen. *Biomedical Signal Processing*. CRC Press, Boca Raton, Florida, USA, 1986.

[20] T. F. Collura. History and evolution of electroencephalographic instruments and techniques. *Journal of clinical neurophysiology*, 10(4):476–504, October 1993.

[21] M. Colombo. *Advances in Interior Point Methods for Large Scale Linear Programming*. PhD thesis, University of Edinburgh, 2007.

[22] Inc. CVX Research. CVX: Matlab software for disciplined convex programming, version 2.0. `http://cvxr.com/cvx`, August 2012.

[23] G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Jersey, 1963.

[24] G. B. Dantzig and M. N. Thapa. *Linear Programming, 1: Introduction*. Springer-Verlag, Inc., New York, 1997.

[25] S. Dehuri, A. K. Jagadev, and S. B. Cho. Epileptic seizure identification from electroencephalography signal using DE-RBFNs ensemble. *Procedia Computer Science*, 23:84–95, 2013. ISSN 1877-0509. doi: http://dx.doi.org/10.1016/j.procs.2013.10.012. 4th International Conference on Computational Systems-Biology and Bioinformatics, CSBio2013.

[26] V. F. Dem'janov, V. N. Malozemov, and D. Louvish. *Introduction to Minimax*. Wiley, 1974. URL `https://books.google.com.au/books?id=sag3nwEACAAJ`.

[27] V. L. Dorr, M. Caparos, F. Wendling, J. P. Vignal, and D. Wolf. Extraction of reproducible seizure patterns based on EEG scalp correlations. *Biomedical Signal Processing and Control*, 2(3):154 – 162, 2007. ISSN 1746-8094. doi: http://dx.doi.org/10.1016/j.bspc.2007.07.002.

[28] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.

[29] Epilepsy Australia Ltd. Epilepsy Australia–Information. `http://www.epilepsyaustralia.net/epilepsy-explained/`, 2014.

[30] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.10.010. URL `http://dx.doi.org/10.1016/j.patrec.2005.10.010`.

[31] S. Ghosh-Dastidar, H. Adeli, and N. Dadmehr. Mixed-band wavelet-chaos-neural network methodology for epilepsy and epileptic seizure detection. *Biomedical Engineering, IEEE Transactions on*, 54(9):1545–1551, Sept 2007. ISSN 0018-9294. doi: 10.1109/TBME.2007.891945.

[32] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 1996.

[33] M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph\_dcp.html`.

[34] M. C. Grant and A. P. Boyd. *Recent Advances in Learning and Control*, chapter Graph Implementations for Nonsmooth Convex Programs, pages 95–110. Springer London, London, 2008. ISBN 978-1-84800-155-8. doi: 10.1007/978-1-84800-155-8_7. URL `http://dx.doi.org/10.1007/978-1-84800-155-8_7`.

[35] C. Guerrero-Mosquera, A. Navia-Vazquez, and A. M. Trigueros. *EEG signal processing for epilepsy*. INTECH Open Access Publisher, 2012.

[36] L. Guo, D. Rivero, J. A. Seoane, and A. Pazos. Classification of EEG signals using relative wavelet energy and artificial neural networks. In *Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary*

Computation, GEC '09, pages 177–184, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-326-6. URL `http://doi.acm.org/10.1145/1543834.1543860`.

[37] L. Guo, D. Rivero, J. Dorado, J. R. Rabual, and A. Pazos. Automatic epileptic seizure detection in EEGs based on line length feature and artificial neural networks. *Journal of Neuroscience Methods*, 191(1):101–109, 2010. ISSN 0165-0270. doi: http://dx.doi.org/10.1016/j.jneumeth.2010.05.020.

[38] L. Guo, D. Rivero, and A. Pazos. Epileptic seizure detection using multiwavelet transform based approximate entropy and artificial neural networks. *Journal of Neuroscience Methods*, 193(1):156–163, 2010. ISSN 0165-0270. doi: http://dx.doi.org/10.1016/j.jneumeth.2010.08.030.

[39] Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL `http://www.gurobi.com`.

[40] D. Henry, D. Sauter, and O. Caspary. Comparison of detection methods: application to k-complex detection in sleep eeg. In *Engineering in Medicine and Biology Society, 1994. Engineering Advances: New Opportunities for Biomedical Engineers. Proceedings of the 16th Annual International Conference of the IEEE*, volume 2, pages 1218–1219, 1994.

[41] C. Iber, S. Ancoli-Israel, A. L. Chesson, and S. F. Quani. The aasm manual for the scoring of sleep and associated events: Rules, technology and technical specifications. Technical report, American Academy of Sleep Medicine, Westchester, IL, 2007.

[42] B. H. Jansen. Artificial neural nets for k-complex detection. *Engineering in Medicine and Biology Magazine, IEEE*, 9(3):50–52, 1990.

[43] B. H. Jansen and P. R. Desai. K-complex detection using multi-layer perceptrons and recurrent networks. *International Journal of Bio-Medical Computing*, 37(3):249–257, 1994.

[44] H. Jeffreys and BS. Jeffreys. Weierstrasss theorem on approximation by polynomials and extension of weierstrasss approximation theory. *Methods of Mathematical Physics*, (3):446–448, 1988.

[45] A. Kales and A. Rechtschaffen. *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects*. U.S. National Institute of Neurological Diseases and Blindness, MD, USA, 1968.

[46] N. Kannathal, U. Rajendra Acharya, C. M. Lim, and P. K. Sadasivan. Characterization of EEG a comparative study.

[47] N. Kannathal, M. L. Choo, U. R. Acharya, and P. K. Sadasivan. Entropies for detection of epilepsy in EEG. *Computer Methods and Programs in Biomedicine*, 80(3):187–194, 2005. ISSN 0169-2607. doi: http://dx.doi.org/10.1016/j.cmpb.2005.06.012.

[48] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, December 1984.

[49] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979.

[50] Y.U. Khan and J. Gotman. Wavelet based automatic seizure detection in intracerebral electroencephalogram. *Clinical Neurophysiology*, 114(5): 898 – 908, 2003. ISSN 1388-2457. doi: http://dx.doi.org/10.1016/S1388-2457(03)00035-X.

[51] V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175. Academic Press Inc., New York, 1972.

[52] M. Kryger, T. Roth, and W. Dement. *Principles and Practice of Sleep Medicine*. Elsevier Saunders, 2005.

[53] C. L. Lawson and R. J. Hanson. *Solving least squares problems*, volume 15 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1995.

[54] S. F. Liang, H. C. Wang, and W. L. Chang. Combination of EEG complexity and spectral analysis for epilepsy diagnosis and seizure detection. *EURASIP J. Adv. Signal Process*, 2010:62:1–62:15, February 2010. ISSN 1110-8657. URL `http://dx.doi.org/10.1155/2010/853434`.

[55] F. Lotte. *Study of Electroencephalographic Signal Processing and Classification Techniques towards the use of Brain-Computer Interfaces in Virtual Reality Applications*. PhD thesis, INSA de Rennes, Dec 2008. URL `https://tel.archives-ouvertes.fr/tel-00356346`.

[56] G. Meinardus, G. Nürnberger, M. Sommer, and H. Strauss. Algorithms for piecewise polynomials and splines with free knots. *Mathematics of Computation*, 53:235–247, 1989.

[57] D. Moloney, N. Sukhorukova, P. Vamplew, J. Ugon, G. Li, G. Beliakov, C. Philippe, H. Amiel, and A. Ugon. Detecting k-complexes for sleep stage identification using nonsmooth optimization. *The ANZIAM Journal*, 52: 319–332, 3 2011.

[58] B. Ofoghi P. Vamplew M. Saleem L. Ma A. Ugon J. Ugon N. Muecke H. Amiel C. Philippe A. BaniMustafa S. Huda M. Bertoli P. Levy J. G. Ganascia N. Sukhorukova, A. Stranieri. Automatic sleep stage identification: difficulties and possible solutions. In A. Maeder and D. Hansen, editors, *Fourth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2010)*, volume 108 of *CRPIT*, pages 39–44, Brisbane, Australia, 2010. ACS. URL `http://crpit.com/confpapers/CRPITV108Sukhorukova.pdf`.

[59] T. Netoff, Y. Park, and K. Parhi. Seizure prediction using cost-sensitive support vector machine. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 3322–3325, Sept 2009. doi: 10.1109/IEMBS.2009.5333711.

[60] V. P. Nigam and D. Graupe. A neural-network-based detection of epilepsy. *Neurological research*, 26(1):55–60, Jan 2004. ISSN 0161-6412. doi: 10.1179/016164104773026534.

[61] J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, Inc., New York, 2006.

[62] R. B. Northrop. *Signals and Systems Analysis in Biomedical Engineering*. CRC Press, Boca Raton, Florida, USA, 2003.

[63] G. Nürnberger. *Approximation by Spline Functions*. Springer-Verlag, Berlin Heidelberg, 1989. URL `http://books.google.com.au/books?id=-0F4QgAACAAJ`.

[64] G. Nürnberger, L. Schumaker, M. Sommer, and H. Strauss. Uniform approximation by generalized splines with free knots. *Journal of Approximation Theory*, 59(2):150–169, 1989. ISSN 0021-9045. doi: http://dx.doi.org/10.1016/0021-9045(89)90150-0.

[65] H. Ocak. Optimal classification of epileptic seizures in EEG using wavelet analysis and genetic algorithm. *Signal Processing*, 88(7):1858 – 1867, 2008. ISSN 0165-1684. doi: http://dx.doi.org/10.1016/j.sigpro.2008.01.026.

[66] H. Ocak. Automatic detection of epileptic seizures in EEG using discrete wavelet transform and approximate entropy. *Expert Systems with Applications*, 36(2, Part 1):2027–2036, 2009. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2007.12.065.

[67] R. B. Pachori and S. Patidar. Epileptic seizure classification in EEG signals using second-order difference plot of intrinsic mode functions. *Computer Methods and Programs in Biomedicine*, 113(2):494–502, February 2014. ISSN 0169-2607. doi: 10.1016/j.cmpb.2013.11.014. URL `http://dx.doi.org/10.1016/j.cmpb.2013.11.014`.

[68] R. Panda, P. S. Khobragade, P. D. Jambhule, S. N. Jengthe, P. R. Pal, and T. K. Gandhi. Classification of eeg signal using wavelet transform and support vector machine for epileptic seizure diction. In *Systems in Medicine and Biology (ICSMB), 2010 International Conference on*, pages 405–408, Dec 2010. doi: 10.1109/ICSMB.2010.5735413.

[69] K. Polat and S. Gne. Classification of epileptiform EEG using a hybrid system based on decision tree classifier and fast fourier transform. *Applied Mathematics and Computation*, 187(2):1017 – 1026, 2007. ISSN 0096-3003. doi: http://dx.doi.org/10.1016/j.amc.2006.09.022.

[70] M. J. D. Powell. *Curve fitting by cubic splines*. Rep. TP 307, Atomic Energy Res. Est., Harwell, England, 1967.

[71] D. K. Ravish, S. Shenbaga Devi, S. G. Krishnamoorthy, and M. R. Karthikeyan. Detection of epileptic seizure in EEG recordings by spectral method and statistical analysis. *Journal of Applied Sciences*, 13(2):207–219, 2013. ISSN 1812-5654. doi: 10.3923/jas.2013.207.219. URL `http://scialert.net/abstract/?doi=jas.2013.207.219`.

[72] J. R. Rice. *The approximation of functions*, volume II. Addison-Wesley, Reading, Massachusetts, 1969.

[73] A. C. Da Rosa, B. Kemp, T. Paiva, F. H. Lopes da Silva, and H. A. C. Kamphuisen. A model-based detector of vertex waves and k complexes

in sleep electroencephalogram. *Electroencephalography and clinical neurophysiology*, 78(1):71–79, 1991.

[74] S. Sanei and J. A. Chambers. *EEG Signal Processing*. Wiley, 2013. ISBN 9781118691236. URL https://books.google.com.au/books?id=f44hLefOz6UC.

[75] S. Santaniello, S. P. Burns, A. J. Golby, J. M. Singer, W. S. Anderson, and S. V. Sarma. Quickest detection of drug-resistant seizures: An optimal control approach. *Epilepsy and Behavior*, 22:S49 – S60, 2011. ISSN 1525-5050. doi: http://dx.doi.org/10.1016/j.yebeh.2011.08.041.

[76] L. Schumaker. Uniform approximation by chebyshev spline functions. II: free knots. *SIAM Journal of Numerical Analysis*, 5:647–656, 1968.

[77] Y. Shang. *Global Search Methods For Solving Nonlinear Optimization Problems*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1997.

[78] Siuly. *Analysis and Classification of EEG signals*. PhD thesis, University of Southern Queensland, Australia, July .

[79] Y. Song and P. Lio. A new approach for epileptic seizure detection: sample entropy based feature extraction and extreme learning machine. *Journal of Biomedical Science and Engineering*, 3(6):556–567, June 2010. doi: 10.4236/jbise.2010.36078. URL http://www.SciRP.org/journal/jbise/.

[80] V. Srinivasan, C. Eswaran, and N. Sriraam. Artificial neural network based epileptic detection using time-domain and frequency-domain features. *Journal of Medical Systems*, 29(6):647–660, 2005. ISSN 0148-5598. doi: 10.1007/s10916-005-6133-1. URL http://dx.doi.org/10.1007/s10916-005-6133-1.

[81] E. Stiefel. Note on jordan elimination, linear programming and tchebycheff approximation. *Numerische Mathematik*, 2(1):1–17. ISSN 0945-3245. doi: 10.1007/BF01386203. URL http://dx.doi.org/10.1007/BF01386203.

[82] A. Subasi. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Systems with Applications*, 32(4):1084–1093, 2007. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2006.02.005.

[83] N. Sukhorukova and J. Ugon. Characterization theorem for best linear spline approximation with free knots. *Dynamics of Continuous, Discrete & Impulsive Systems*, 17(5):687–708, 2010.

[84] N. Sukhorukova and J. Ugon. Characterization theorem for best polynomial spline approximation with free knots. *Submitted*, 2013.

[85] N. Sukhorukova and J. Ugon. Chebyshev approximation by linear combinations of fixed knot polynomial splines with weighting functions. *Journal of Optimization Theory and Applications*, pages 1–14, 2016. ISSN 1573-2878. doi: 10.1007/s10957-016-0887-0. URL http://dx.doi.

org/10.1007/s10957-016-0887-0.

[86] Y. Tang and D.M. Durand. A tunable support vector machine assembly classifier for epileptic seizure detection. *Expert Systems with Applications*, 39(4):3925 – 3938, 2012. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2011.08.088.

[87] Z. Tang and N. Ishii. Detection of the k-complex using a new method of recognizing waveform based on the discrete wavelet transform. Technical Report 1, The Institute of Electronics, Information and Communication Engineers (IEICE), 1995.

[88] K. C. Toh, M. J. Todd, and R. H. Tutuncu. SDPT3 version 4.0 – a matlab software for semidefinite-quadratic-linear programming. http://www.math.nus.edu.sg/~mattohkc/sdpt3.html.

[89] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. Miscellaneous Bks. Cambridge University Press, 1997.

[90] R. H. Tütüncü, K. C. Toh, and M. J. Todd. SDPT3 – a matlab software package for semidefinite-quadratic-linear programming, version 3.0. http://www.math.cmu.edu/~reha/Pss/guide3.0.pdf, August 2001.

[91] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis. Automatic seizure detection based on time-frequency analysis and artificial neural networks. *Computational Intelligence and Neuroscience*, 2007:80510, 2007. ISSN 1687-5265. doi: 10.1155/2007/80510. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2246039/.

[92] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis. Epileptic seizure detection in EEGs using time frequency analysis. *Information Technology in Biomedicine, IEEE Transactions on*, 13(5):703–710, Sept 2009. ISSN 1089-7771. doi: 10.1109/TITB.2009.2017939.

[93] H. Q. Vu, G. Li, N. S. Sukhorukova, G. Beliakov, S. Liu, C. Philippe, H. Amiel, and A. Ugon. K-complex detection using a hybrid-synergic machine learning method. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(6):1478–1490, Nov 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2012.2191775.

[94] Weka web site. www.cs.waikato.ac.nz/ml/weka/.

[95] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, 2nd edition, 2005.

[96] S. Wold. Spline functions in data analysis. *Technometrics*, 16(1):pp. 1–11, 1974. ISSN 00401706. URL http://www.jstor.org/stable/1267485.

[97] M. H. Wright. Interior methods for constrained optimization. *Acta Numerica*, 1:341–407, 1 1992. ISSN 1474-0508. doi: 10.1017/S0962492900002300. URL http://journals.cambridge.org/article_S0962492900002300.

[98] S. Xie and S. Krishnan. Wavelet-based sparse functional linear model with

applications to eegs seizure detection and epilepsy diagnosis. *Medical and Biological Engineering and Computing*, 51(1-2):49–60, 2013. ISSN 0140-0118. doi: 10.1007/s11517-012-0967-8. URL http://dx.doi.org/10.1007/s11517-012-0967-8.

[99] D. Yang, G. D. Peterson, H. Li, and J. Sun. An fpga implementation for solving least square problem. In *17th IEEE Symposium on Field Programmable Custom Computing Machines, FCCM'09*, pages 303–306, 2009.

[100] Z. Roshan Zamir, N. Sukhorukova, H. Amiel, A. Ugon, and C. Philippe. Optimization-based features extraction for k-complex detection. In Mark Nelson, Tara Hamilton, Michael Jennings, and Judith Bunder, editors, *Proceedings of the 11th Biennial Engineering Mathematics and Applications Conference, EMAC-2013*, volume 55 of *ANZIAM J.*, pages C384–C398, August 2014. http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/7802 [August 27, 2014].

[101] Z. Roshan Zamir, N. Sukhorukova, H. Amiel, A. Ugon, and C. Philippe. Convex optimisation-based methods for k-complex detection. *Applied Mathematics and Computation*, 268:947 – 956, 2015. ISSN 0096-3003. doi: http://dx.doi.org/10.1016/j.amc.2015.07.005.

[102] Y. Zhang, G. Zhou, Q. Zhao, J. Jin, X. Wang, and A. Cichocki. Spatial-temporal discriminant analysis for ERP-based brain-computer interface. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 21 (2):233–243, March 2013. ISSN 1534-4320. doi: 10.1109/TNSRE.2013.2243471.

[103] W. Zhou, Y. Liu, Q. Yuan, and X. Li. Epileptic seizure detection using lacunarity and bayesian linear discriminant analysis in intracranial eeg. *Biomedical Engineering, IEEE Transactions on*, 60(12):3375–3381, Dec 2013. ISSN 0018-9294. doi: 10.1109/TBME.2013.2254486.

[104] ZIENA. www.artelys.com/en/optimization-tools/knitro.

Appendix A

Development of MeanFreq as a Classifier

The numerical experiments of Chapter 5 demonstrate that the frequency (ω) is an essential feature for automatic detection of K-complexes. Therefore we kept the frequency as the only feature of an EEG signal instead of $m_1 n + 1$ and $2m_1 n + 2$ features of LLSOM1, UOM1 and LLSOM2, UOM2 respectively. Table A.1 presents the classification accuracies of K-complexes after optimization-based preprocessing approaches (LLSOM1, LLSOM2, UOM1 and UOM2) based on the frequency (ω, one feature). The important observation is that the classification accuracies are consistent in each of optimization methods after feature extraction. The performance of all classifiers is similar to each other. The most accurate optimization method is LLSOM2 regardless of the number of essential features extracted. These facts encourage us to develop a classifier called “MeanFreq” based on the mean frequency for K-complexes and non-K-complexes.

First, this classifier evaluates the mean frequency for both classes (K-complex and non-K-complex) on the training set segments. Second, it calculates the threshold value for the mean frequency (TMF) as the average of the mean frequencies for K-complexes and non-K-complexes (see Table A.2, TMF). Finally, it assigns all the test set segments with the frequency less than or equal to the calculated threshold to the class with K-complexes, otherwise to the class without K-complexes. Despite the simplicity of “MeanFreq”, it produces very reasonable results, especially in the case of UOM1 with the accuracy of 74%.

Table A.1: Classification accuracy on the test set for the preprocessed dataset, one feature (ω).

Classifiers	LLSOM1	LLSOM2	UOM1	UOM2
LibSVM	74%	79%	53%	79%
Logistic	74%	79%	53%	74%
RBF Network	74%	79%	53%	74%
SMO	74%	79%	53%	74%
LazyIB1	74%	79%	53%	74%
LazyIB5	74%	79%	53%	79%
KStar	74%	79%	53%	74%
LWL	74%	79%	53%	74%
OneR	74%	79%	53%	74%
J48	74%	79%	47%	74%
J48graft	74%	79%	47%	74%
LMT	74%	79%	53%	79%
MeanFreq	68%	68%	74%	58%

Table A.2: Numerical results on optimization-based preprocessing.

Optimisation Models	**LLSOM1**	**LLSOM2**	**UOM1**	**UOM2**
CPU time (in seconds)	21	346	2654	4270
MFK [a]	1.1000	1.1000	1.1000	2.1323
MFNK [b]	1.6385	1.7667	2.2538	3.5615
TMF [c]	1.4810	1.5167	1.8286	3.0190

[a] Mean Frequency for K-complexes (all instances).

[b] Mean Frequency for non-K-complexes (all instances).

[c] Threshold value for the mean frequency (on the training set).